INTEGRAL Science Data Centre

SPI Analysis User Manual

Reference : ISDC/OSA-UM-SPI
Issue : 9.0
Date : 1 March 2010

INTEGRAL Science Data Centre
Chemin d'Écogia 16
CH–1290 Versoix
Switzerland

http://isdc.unige.ch

Authors and Approvals

ISDC	SPI Analysis User Manual	
1 March 2010	9.0	

Prepared by : P. Dubath
I. Kreykenbohm
C. Ferrigno

Agreed by : R. Walter .....

Approved by : T. Courvoisier .....

Document Status Sheet

ISDC	SPI Analysis User Manual	
2 April 2003	1.0	First Release.
19 May 2003	1.1	Update of the First Release. Sections 7.2, 7.3, 7.5.2, 9, 8.3 and Table 4 were updated.
18 July 2003	2.0	Second Release. Sections 6, 8.9, 10 and the bibliography were updated.
8 December 2003	3.0	Third Release. Sections 6, 10 and the bibliography were updated.
19 July 2004	4.0	Forth Release. Sections 5, 6, 8, 10 and the bibliography were updated.
6 December 2004	4.2	Update of Forth Release. Sections 3.3, 5, 6, 10 and the bibliography were updated.
31 May 2005	5.0	Major update for OSA 5.0 Release. All sections were modified.
31 June 2006	6.0	Major update for OSA 6.0 Release. Cookbook adapted to the new spi_science_analysis, with added functionality on phase resolve analyses and selection of good pointing.
10 September 2007	7.0	Update for OSA 7.0 Release. Cookbook adapted to the new features of spi_science_analysis.
1 June 2009	8.0	Update for OSA 8.0 Release. Cookbook updated for the introduction of spimodfit_analysis.
1 March 2010	9.0	Update for OSA 9.0 Release. Revision of flat-field options.
28 APR 2010	Printed	

Contents

Acronyms and Abbreviations x

Glossary of Terms xi

1 Introduction 1

I Instrument Definition 2

2 Scientific Performance Summary 2

3 Instrument Description 3

3.1 The Overall Design 3

3.2 The Passive Mask 4

3.3 The Detectors 4

3.4 Event Types 5

3.5 Pulse Shape Discriminator (PSD) 6

3.6 Anti-Coincidence Subassembly (ACS) 6

3.7 The Plastic Scintillator Anti Coincidence Subassembly (PSAC) 6

3.8 Electronics 7

3.9 Operating Modes 7

3.10 Dithering Strategy 7

4 Performance of the Instrument 8

4.1 Components and Sources of Instrumental Background 8

4.1.1 Continuum Background 9

4.1.2 511 keV Background 9

4.1.3 Background Gamma-Ray Lines 10

4.2 Instrumental Characterization and Calibration 10

4.3 Measured Performance 10

4.3.1 Imaging Resolution 10

4.3.2 Spectral Resolution 11

4.3.3 Sensitivity 11

4.3.4 Dithering Sensitivity Degradation 11

4.3.5 Detection of off-axis-sources 13

4.3.6 Imaging Capabilities 13

4.3.7 Timing Capabilities 13

II Data Analysis **14**

5 Overview . . . 14

6 Cookbook . . . 15

6.1 Setting up the analysis environment . . . 15

6.1.1 Input data organization . . . 15

6.1.2 Downloading data from the ISDC archive . . . 16

6.1.3 Setting the software environment . . . 17

6.2 Observation group (OG) creation . . . 18

6.3 Image reconstruction without input catalogue . . . 18

6.3.1 Data screening . . . 28

6.4 Image reconstruction with catalogue information . . . 30

6.4.1 Creating a new source_cat.fits file . . . 30

6.4.2 Using a source_cat.fits file in the analysis . . . 31

6.5 Spectral Extraction . . . 32

6.6 Light-curves . . . 33

6.7 Spectral Extraction using *spimodfit* . . . 35

6.7.1 Further settings . . . 36

6.8 Phase Resolved Analysis . . . 37

6.8.1 Input ephemeris file . . . 41

6.9 Gamma Ray Burst imaging . . . 41

6.10 Gamma Ray Burst spectroscopy . . . 46

7 Tips and Tricks . . . 50

7.1 Example of command-line run . . . 50

7.2 Repeating and saving analyses . . . 52

7.3 Pseudo-detectors and detector list . . . 52

7.4 Catalogue extraction . . . 53

7.5 Event energy correction . . . 53

7.6 Event binning, GTI and dead times . . . 53

7.7 Background modeling . . . 54

7.8 Image reconstruction . . . 54

7.9 Biases from bright sources “just outside the image FOV” . . . 55

7.10 Spectral extraction . . . 55

7.11 Variable sources and timing analysis . . . 55

7.12 Converging on a good solution - checking residuals . . . 56
7.13 High time resolution with SPI . . . 56
7.14 SPI failed detectors . . . 56
7.15 Annealing phases . . . 57
7.16 Instrument modes and SPI data . . . 57
7.17 Crowded fields . . . 58
7.18 Fitting SPI spectra using XSPEC . . . 58
7.19 Using a SPIROS output catalogue in input (source_res.fits to source_cat.fits) 59
7.20 Adding a new parameter in a spi_science_analysis analysis . . . 59
7.21 GTIs in SPI data analysis . . . 60
8 Data analysis steps . . . 61
8.1 Catalog source extraction – CAT_EXTRACT . . . 61
8.2 Energy correction – SPI_GAIN_CORR . . . 61
8.3 Pointing definition – SPI_OBS_POINT . . . 62
8.4 Energy bin definition and events binning . . . 62
8.4.1 SPIBOUNDS . . . 62
8.4.2 SPI_OBS_HIST . . . 63
8.4.3 SPI_ADD_SIM . . . 63
8.5 Background model generation – SPI_OBS_BACK . . . 64
8.6 Image reconstruction and spectral extraction – SPIROS . . . 65
8.6.1 IMAGING mode . . . 65
8.6.2 SPECTRAL extraction mode . . . 66
8.6.3 TIMING analysis mode . . . 66
8.6.4 Parameter list . . . 67
8.7 Spectral Extraction – SPMODFIT . . . 69
8.7.1 Use of *spimodfit* . . . 70
8.7.2 spimodfit_analysis script . . . 70
8.7.3 Use and syntax of the parameter file . . . 71
8.7.4 Suggested settings for point source analysis . . . 72
8.7.5 Known issues . . . 73
8.8 Response interpolation – SPIRMF . . . 73
8.9 Image reconstruction – SPISKYMAX . . . 74
8.10 Cleaning Tool - SPI_CLEAN . . . 74

9 Additional tools 76
9.1 SPI_SPECTRAL_DISPLAY 76
9.2 SPI_DSP_DISPLAY 77
10 Known limitations 78

List of Figures

1 A cut-away view of the SPI instrument . . . 3
2 Numbering of SPI detectors. . . . 4
3 The mask of the SPI instrument . . . 5
4 Dithering patterns and instrument field-of-views . . . 8
5 The SPI continuum background components . . . 9
6 Spectrum example . . . 11
7 Energy resolution . . . 12
8 Continuum sensitivity . . . 12
9 Narrow line sensitivity . . . 13
10 spi_science_analysis main GUI . . . 19
11 Energy bin definition GUI . . . 20
12 Main background GUI . . . 21
13 Options for the flat field background . . . 21
14 Options for the template background . . . 23
15 Options for modeling the background using tracers like GEDSAT . . . 24
16 SPIROS main GUI . . . 25
17 SPIROS imaging mode GUI . . . 26
18 Catalogue extraction GUI . . . 30
19 Crab spectrum from cookbook example . . . 34
20 SPIROS timing mode GUI . . . 34
21 Crab light-curve from the cookbook example . . . 35
22 GUI of `spimodfit_analysis` with the default setting. . . . 36
23 GUI of `spimodfit_analysis` energy definition, here 22 equally spaced logaritmic bins have been selected. . . . 37
24 GUI of `spimodfit_analysis` regarding *spimodfit* . . . 37
25 GUI of `spimodfit_analysis` regarding the *spimodfit* background multipliers. . . . 38
26 Count rate spectrum of the Crab from the cookbook example and the residuals from the best fit model. . . . 38
27 Histogram options GUI . . . 39
28 Phase parameter GUI . . . 39
29 Orbit parameter GUI . . . 40
30 Illustration of the different GRB start/stop times . . . 42
31 Output of og_create on the terminal . . . 42

32 Main GUI window of the spi_science_analysis script 42
33 Energy boundaries (or ebounds) GUI 43
34 Background GUI 43
35 spiros GUI main window 44
36 spiros GUI imaging window 44
37 Outputs of spi_grb_analysis when invoked without any input parameters 45
38 Main GUI window of the spi_science_analysis script 47
39 Energy boundaries (or ebounds) GUI 47
40 spiros GUI main window 48
41 Outputs of spi_grb_analysis when invoked without any input parameters 49
42 GUI with the standard settings for a flat-field background and logarithmically increasing energy binning. 71
43 GUI with the standard settings for internal background computation, no image fitting and standard background multipliers. 71

List of Tables

1 Main characteristics of the SPI instrument 2
2 511 keV background line strength with and without PSAC. 10
3 Overview of the successive steps of the spi_science_analysis script 14
4 SPI detector failures 57
5 SPI annealing periods 58
6 *cat_extract* parameters included in the main script 61
7 *spibounds* parameters included into the main script 62
8 *spi_add_sim* parameters included into the main script 63
9 *spi_obs_back* parameters included into the main script 64
10 *spiros* parameters included into the main script. 67
11 *spi_clean* parameters 75
12 *spi_spectral_display* parameters 76
13 *spi_dsp_display* parameters 77

Acronyms and Abbreviations

ACS	Anti-Coincidence Shield
ARF	Ancillary Response Files
BGO	Bismuth Germanate
DFEE	Digital Front End Electronics
DOL	Data Object Locator
DPE	Data Processing Electronics
DS	Data Structure
FCFOV	Fully Coded Field of View
FOV	Field of View
FWHM	Full Width at Half Maximum
GRB	Gamma Ray Burst
GTI	Good Time Interval
GUI	Graphic User Interface
IC	Instrument Characteristics
IJD	ISDC Julian Date
IRF	Image Response File
ISDC	Integral Science Data Center
ISOC	Integral Science Operations Centre
OBT	On-Board Time
OG	Observation Group
PHA	Pulse Height Amplifier
PI	Pulse Invariant
PSA	Pulse Shape Amplifier
PSAC	Plastic Scintillator Anti Coincidence Subassembly
PSD	Pulse Shape Discriminator System
RMF	Redistribution Matrix Files
ScW	Science Window
SWG	Science Window Group
TBW	To be written
TM	Telemetry
UTC	Coordinated Universal Time

Glossary of Terms

- *ISDC system:* the complete ground software system devoted to the processing of the *INTEGRAL* data and running at the ISDC. It includes contributions from the ISDC and from the *INTEGRAL* instrument teams.
- *Pointing:* Period during which the spacecraft axis pointing direction remains stable. Because of the *INTEGRAL* dithering strategy, the nominal pointing duration is of order of 30 minutes.
- *Slew:* Period during which the spacecraft is manoeuvered from one stable position to another, i.e., from one pointing to another.
- *Science Window (ScW):* For the operations, ISDC defines atomic bits of *INTEGRAL* operations as either a pointing or a slew, and calls them ScWs. A set of data produced during a ScW is a basic piece of *INTEGRAL* data in the ISDC system.
- *Observation:* Any group of ScW used in the data analysis. The *observation* defined from ISOC in relation with the proposal is only one example of possible *ISDC observations*. Other combinations of Science Windows, *i.e.,* of observations, are used for example for the Quick-Look Analysis, or for Off-Line Scientific Analysis.

1 Introduction

The present 'SPI Analysis User Manual' includes two major parts, (1) a description of the INTEGRAL SPI instrument, and (2) information about the SPI data analysis system provided by the ISDC. A more general overview on the *INTEGRAL* Data Analysis can be found in the 'Introduction to the *INTEGRAL* Data Analysis' [1].

People starting with SPI data analysis are expected to skip the first part of the document as they should already have some basic knowledge about the instrument. This part can always be consulted later. A large part of these information provided duplicates those found in the ISOC AO documentation[3].

We recommend to start with the "overview" and "cookbook" in Sect. 5 and 6, respectively. The cookbook is meant to be a "hands on" experience. It guides you through a practical example, and provided some (limited) explanatory information.

Once you have exercised the cookbook example, browse the "Tips and Tricks" section (Sect. 7), and start your own analysis. Come back to the "cookbook" and the "Tips and Tricks" sections when you are confronted with problems or difficult decisions. These two sections should provide enough information for users of the main SPI analysis script, the so-called spi_science_analysis and pimodfit_analysis.

Since the analysis of SPI data can be difficult in crowded fields or when there are variable source, it is important that the user reads carefully the documents provided by the SPI team [10] in order to understand how these situations should be handled. The documents refer to the older pipeline (spi_science_analysis), but the general principles are valid also for the alternative method (spimodfit) delivered with version 8.0 of the OSA package.The documents are available at the page http://www.isdc.unige.ch/integral/archive

For more expert users, who want to develop their own scripts or to execute individual programs independently, we advise to continue with Sect. 8, and most important, to look at the individual program "user manuals". These user manuals can be downloaded from the following WWW page.

http://isdc.unige.ch/Support/spi/doc/doc.html

A thorough scientific validation of an important part of the ISDC software can be found in the paper "The INTEGRAL spectrometer SPI: performance of point-source data analysis" published in MNRAS [4], which also contains a number of useful informations and recommendations.

For analysis of Gamma-ray bursts a dedicated script spi_grb_analysis is available. This script is documented on-line at

http://isdc.unige.ch/index.cgi?Support+spi

Part I

Instrument Definition

2 Scientific Performance Summary

The SPI telescope has been designed to provide at the same time a medium angular resolution and an excellent energy resolution in the range 20 keV – 8 MeV. The main characteristics of the instrument are given in Table 1. The imaging capabilities are limited by the fact that SPI consists of only 19 detectors (pixels), and that the data are strongly background dominated. To help distinguishing between source and background photons, the number of "measurements" is increased by dithering the whole spacecraft around the target of interest. A typical observation is thus split into a number of "pointings" whose attitude differ by about 2 degrees, see Section "Strategies of scientific Observations" in the Introduction to the INTEGRAL Data Analysis [1]).

,

Table 1: Main characteristics of the SPI instrument

Energy range	20 keV – 8 MeV
Energy resolution (FWHM)	2.35 keV at 1.33 MeV
Detector area	$\sim$500cm^2
Field-of-View:	16°$\times$ 16°(fully coded) 35°$\times$ 35°(zero coded)
Angular resolution	2.8° for point sources.
Point source positioning	<1.3° for point sources (depending on point source intensity).
Narrow line Sensitivity	5.$\times 10^{-6}$ ph/cm^2/s at 1 MeV
Timing accuracy	160 μs

3 Instrument Description

3.1 The Overall Design

The SPI instrument is a coded mask spectrometer designed with a hexagonal geometry to maximize its compactness. An overall cut-out view of the instrument is given in Figure 1.

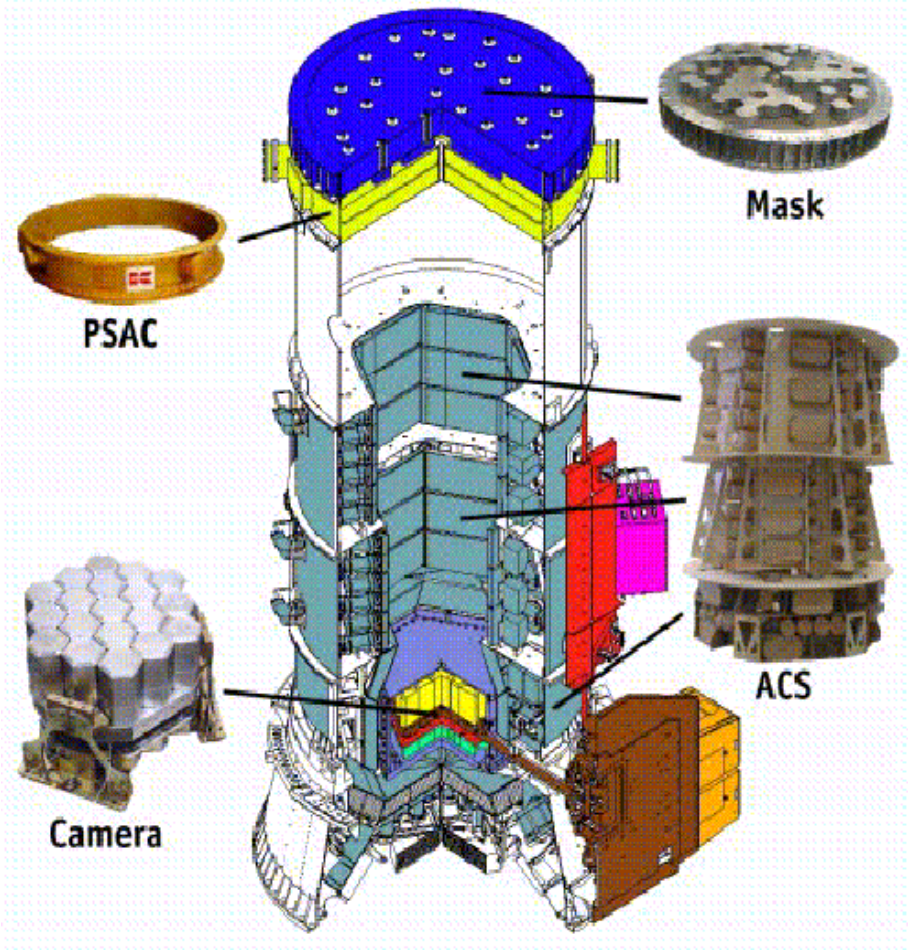


Figure 1: A cut-away view of the SPI instrument. The mask, plastic scintillator, camera and ACS subsystems are highlighted.

The detector plane of the instrument consists of 19 cooled, hexagonal shaped, high purity Germanium detectors, providing a total area of about 500 cm^2. In Figure 2 the detector numbering is shown. An Anti-Coincidence Shield (ACS) defines the instrument Field-Of-View (FOV) by providing veto pulses for photons and particles coming from the sides. A plastic scintillator (PSAC), placed underneath the coded mask detects charged particles coming through the FOV and also provides veto signal. The sensitivity of the instrument is limited by the background due to the primary and secondary cosmic ray particles and the cosmic background radiation.

Each photon that is absorbed in a Germanium detector will give a pulse that is sent to the electronics. The electronics analyze the incoming pulses, reject those simultaneous with veto signals, and tags good photons with the energy, the time and the type of event (see Section 3.4 for the discussion on the possible event types). These data are then sent to the ground. All photons detected by the detectors are summed into background spectra and are sent to the ground every 30 minutes.

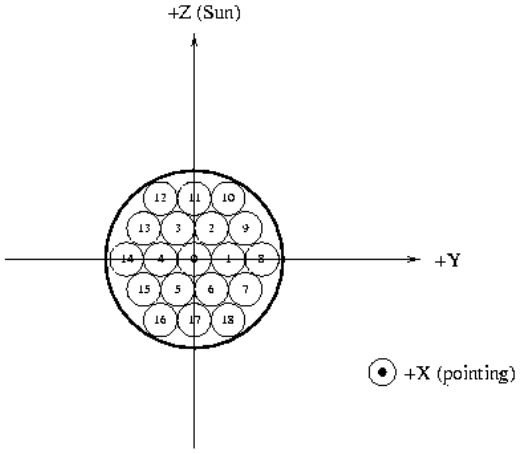


Figure 2: Numbering of SPI detectors.

3.2 The Passive Mask

The passive mask is located on top of the SPI instrument, above the plastic scintillator. The purpose of the mask is to code the incident gamma rays in the field-of-view, providing the instrument imaging capabilities. The mask consists of a pattern made of hexagonal tungsten blocks, with a 120°rotation symmetry.

The mask is made of 127 elements arrange in a 78 cm diameter circle. Of these elements, 63 are opaque and 64 are transparent. Each opaque element is 30 mm thick and 60 mm flat to flat in size. The tungsten elements stop the gamma ray radiation in the range 20 keV to 8 MeV with an absorption efficiency greater than 95% at 1 MeV. The holes in the mask have a gamma ray transparency of 60% at 20 keV and 80% at 50 keV. The mask is located 171 cm from the detector plane. A picture of the mask pattern is given in Figure 3.

3.3 The Detectors

The SPI detectors are the only elements of the INTEGRAL spacecraft that are actively cooled down. A low temperature is required for an optimum sensitivity and resolution, as well as for limiting the radiation damage. An active cooling system brings the temperature of the cold plate on which the detectors are mounted down to 85 K, using two pairs of cryocoolers. In normal operating mode all coolers work simultaneously. In case of failure of one of the cryocoolers, or of the electronics, the instrument can function with only one pair of cryocoolers in a degraded mode, with a detector temperature of about 100 K. The detector assembly is placed inside a cold box, which is kept at approximately 210 K by the passive cooling system. All temperatures of the cryostat subsystems are regularly monitored to provide temperature information that can be used for the data processing.

The detector plane is composed of 19 hexagonal-shaped Germanium detectors, encapsulated into beryllium capsules, and mounted in the most compact configuration on the cold plate. The size of a detector is 5.6 cm, flat-to-flat, with a height of 7 cm. The detectors are mounted with minimum space between them, such that the axes of two adjacent detectors are 6 cm apart.

At the time of writing, two of these 19 detectors have ceased to function properly and are disabled.

- Detector 2 died on 6. December 2003 (2003-12-06) revolution 140.
- Detector 17 died on 18. July 2004 (2004-07-18) Revolution 215.

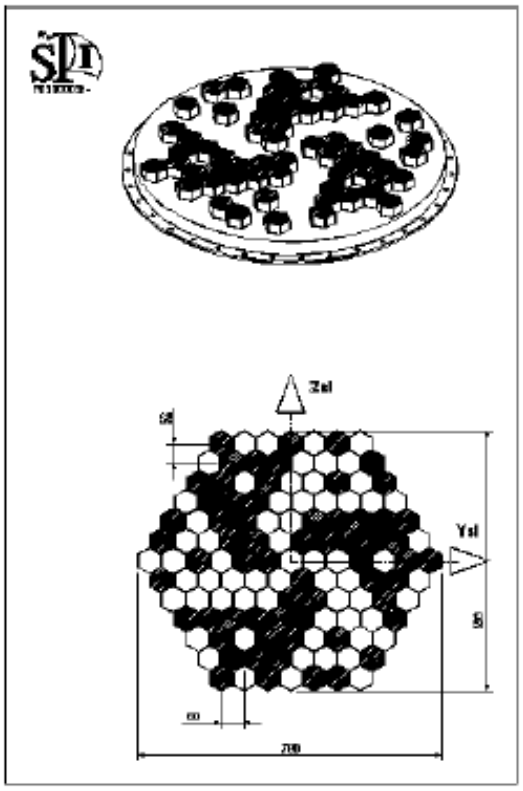


Figure 3: The passive mask of the SPI instrument. The bottom picture indicates the direction of the spacecraft Y and Z axes with respect to the mask.

- Detector 5 died on 19. February 2009 (2009-02-19) Revolution 776.

The death of the detectors results in a decrease of the effective area of the instrument, roughly by 1/19 or 2/19 - from revolution 215 on, the effective area is about 90% of the original area.

In order to recover from the radiation damage to the Germanium crystal structure, every 6 to 12 months, the detectors go through an annealing operation. They are heated to 105°C for a few days. The instrument is not available for scientific observations during the time needed for the annealing operation and the cooling phase afterwards.

The signals from the detectors pre-amplifiers are sent to the amplification chain, which is made up of a Pulse Shape Amplifier and a Pulse Height Amplifier. The pulses are amplified in such a way that the performance of the spectrometer is optimized. This is done by making a compromise between getting the best signal to noise ratio for the pulses, operating in the full 20 keV – 8 MeV energy band of the instrument, and making the output pulses insensitive to the fluctuations in the detector signal rise time.

3.4 Event Types

The photon which entered the telescope can be detected due to its interaction with the absorbing material of the detector. Three major types of interactions play a dominant role: photoelectric absorption, Compton scattering and pair production. In the photoelectric absorption process a photon undergoes an interaction with an absorber atom in which the photon completely disappears. In its place an energetic photo electron is ejected by the atom. The photo electron carries off most of the original photon energy. The Compton scattering takes place between the incident gamma-ray photon and an electron in the absorbing material. The incoming photon is deflected and it

transfers a portion of its energy to the electron. In the pair production process the gamma-ray photon disappears and is replaced by an electron-positron pair. The positron will annihilate in the absorbing medium and two annihilation photons are normally produced as secondary products of the interactions. Depending on the size of the detector and on the energy of the incoming photon, a photon scattered in a Compton interaction can escape the detector, or undergo a second interaction. The pairs of 511 keV photons, produced by the annihilation of the positrons resulting from pair creation, can also produce other interactions or escape the detector.

In SPI the detector array is used to recover the total energy of an incoming photon. This contributes to raise the full-energy peak and to reduce the Compton continuum. Unfortunately it also degrades spatial information as it is not possible, in general, to determine which detector was hit first.

All signals produced from the detection plane are also directed towards the Pulse Shape Discriminator (PSD) System (see more details in Section 3.5).

All events detected by SPI are classified in accordance with how many detectors respond to an incoming photon. The event detected by only one detector is called a *single event*. Multiple events are detected by several detectors and the value of multiplicity corresponds to the number of the detectors hit.

3.5 Pulse Shape Discriminator (PSD)

The photons interacting in a given detector can either deposit all their energy at a single location, the so-called localized events, or in two or more sites, the un-localized events. Pre-launch simulations predicted that the instrumental background would be dominated by localized β^- decays in the energy range 200 keV – 1 MeV.

The PSD electronic sub-system was design to determine whether interactions are localized or not. Unfortunately, the fraction of β^- decays observed in flight data turns out not to be large enough for a significant background reduction. Despite important efforts from SPI experts, it has not been possible to improve significantly the data analysis by using the information provided by the PSD sub-system.

3.6 Anti-Coincidence Subassembly (ACS)

The main function of the Anti-Coincidence Subassembly (ACS) is to shield the Germanium detectors against background (photons and particles) from sources outside the field-of-view. The ACS system consists of 91 Bismuth Germanate scintillator crystals in combination with photo multiplier tubes.

The scintillator crystals are used to convert all incoming events into photons in the 480 nm region (visible light). Photo-multiplier tubes are used to detect these photons and convert them into electrical pulses which are sorted, normalized and summed up by the ACS electronics. Each photon registered by ACS induces a time tagged veto signal directed to the Digital Front End Electronics (DFEE) which reject any simultaneous detector events.

3.7 The Plastic Scintillator Anti Coincidence Subassembly (PSAC)

The purpose of the plastic scintillator subassembly (PSAC) is to reduce the 511 keV background due to particle emission by the passive mask. The detector consists of a plastic scintillator inside a light tight box, located just below the passive mask. It has a good gamma ray transparency, and actively detects particles which deposit energies in excess of 300 keV. The light flashes that are produced by the impacts of these particles are detected by four photo multiplier tubes located around the light-tight box and converted into electrical pulses which are processed by the PSAC

electronics assembly. The electronics produced a veto signal associated with the detected events, which are later merged with the ACS veto signal by the ACS front end electronics.

3.8 Electronics

The SPI on-board electronics is in charge of the real time acquisition, assembly, time stamping and intermediate storage of the various pieces of information coming from the detectors, the PSD, the ACS etc. Events are subdivided into classes depending on their origin in the instrument (detector electronics, detectors, PSD, veto shield) and are used within the overall event energies and system monitoring statistics (dead time, signal counts etc.). The detected events are time tagged with a 20 MHz local clock, which provides the timing resolution. The reset (timing reference) is done with the 8 Hz satellite clock.

3.9 Operating Modes

The SPI instrument has only one mode for normal observations. All scientific observations with SPI are done in so called photon-by-photon mode with a high temporal resolution. In this mode scientific data are collected and transmitted to the ground for each photon.

In case the SPI telemetry is continuously overflowing due to background radiation that is higher than expected or due to a strong solar flare, the instrument can be operated in a degraded science mode (TM emergency mode). In this case, the single detector events are binned on-board, and downloaded in the form of spectra. Thus the time information of each single photon is lost, but the analysis software is still able to handle the data in order to produce spectra and images as described in this document. The maximum data generation rate in this mode will be about half the rate for normal photon by photon mode.

3.10 Dithering Strategy

The SPI data are background dominated. The background in each of the 19 independent detectors can vary in time in a different way. Several types of background variations can be anticipated:

- short-term variations due to solar activity,
- variations over the orbital period (related to the position of INTEGRAL in the orbit).
- long-term variations over the mission duration.

In order to reconstruct the image on the detectors for all pixels in the field-of-view (25 degree field, with 2 degree resolution) for a single pointing a set of 19 equations with 156 unknowns would need to be solved. This is impossible, and the only way to increase the number of equations and make the system solvable is to use more pointings. Thus, in order to solve the problem of background determination an appropriate dithering strategy has to be adopted for every observation.

The imaging performance of SPI depends on the dithering pattern that is used. In general the greater the number of pointings, the better the imaging. It was shown that, when using the hexagonal dither pattern the reconstructed point source response function shows very strong side lobes at distances of 10° to 20° from the center. Therefore this mode should only be used for isolated point sources and is not really suitable for imaging. The side lobes are still present, but significantly less with the 5 by 5 dither pattern (about 50% of the hexagonal case). To remove these side lobes, which will cause artifacts in reconstructed images, the only possibility is to enlarge the imaged area by using many more pointings (i.e. many dithering patterns).

Schematic view of possible dithering patterns is shown in Figure 4. A different dithering strategy can be adopted depending on the circumstances. For SPI, a 5 by 5 dithering pattern with 2° spacing around the target source are almost always preferable. In the case, there is only a bright, isolated source in the field-of-view, an hexagonal dithering pattern, with one pointing centered on the source, and surrounded by six pointings with distances of 2°, is also acceptable.

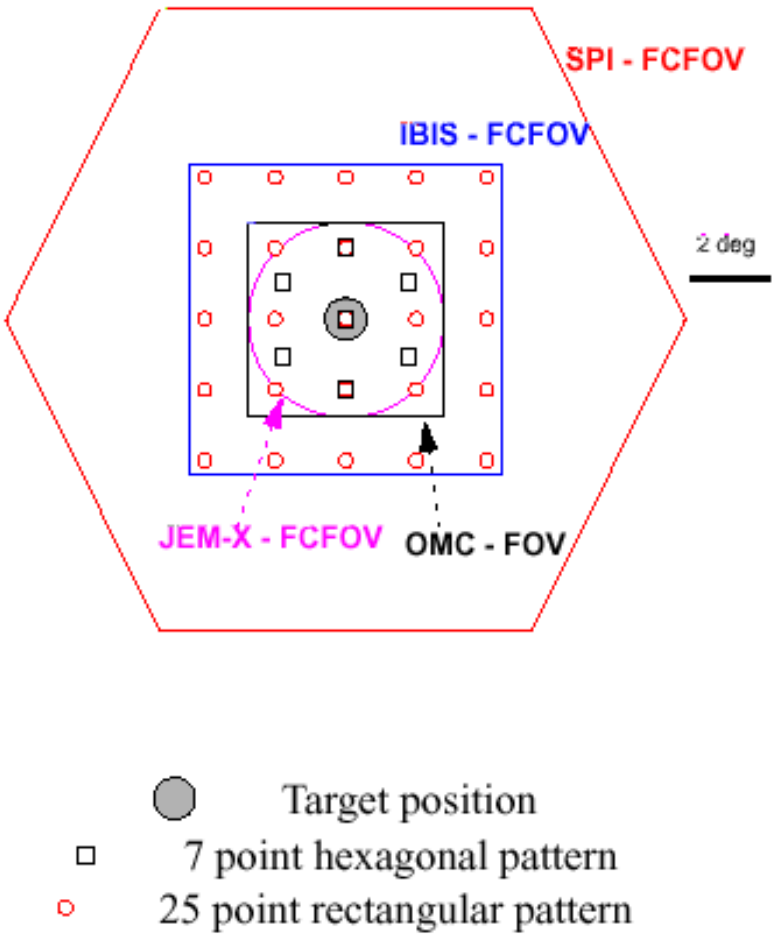


Figure 4: Schematic view of the dithering patterns and of the instrument Fully Coded Fields of Views.

Both dithering patterns use a dwell time of the order of 30 minutes per point.

4 Performance of the Instrument

4.1 Components and Sources of Instrumental Background

As we have already mentioned above, the SPI instrument is background limited. The sensitivity of the instrument is therefore largely dependent on the background and on the correct identification of background photons. The background can be divided into the following main components:

- continuum radiation
- 511 keV line radiation

- gamma ray lines

4.1.1 Continuum Background

The continuum background can be split into several components, depending on their origin. First the radiation coming from outside the instrument. This can be the cosmic diffuse gamma ray flux that comes in through the instrument aperture, or leakage through the BGO shield of cosmic diffuse gamma ray radiation and gamma continuum radiation from the spacecraft (induced by energetic cosmic ray particles). Secondly scattering in the germanium detectors of neutrons that were produced in the spacecraft or other parts of the instrument. Thirdly background components produced inside the spectrometer detectors. These consist of localized β^- decays, non localized β^- decays and β^+decays. The continuum emission from the mask and the BGO emission when the veto electronics are blacked out (veto dead time) are negligible. The individual components and the total continuum background emission are illustrated in Figure 5.

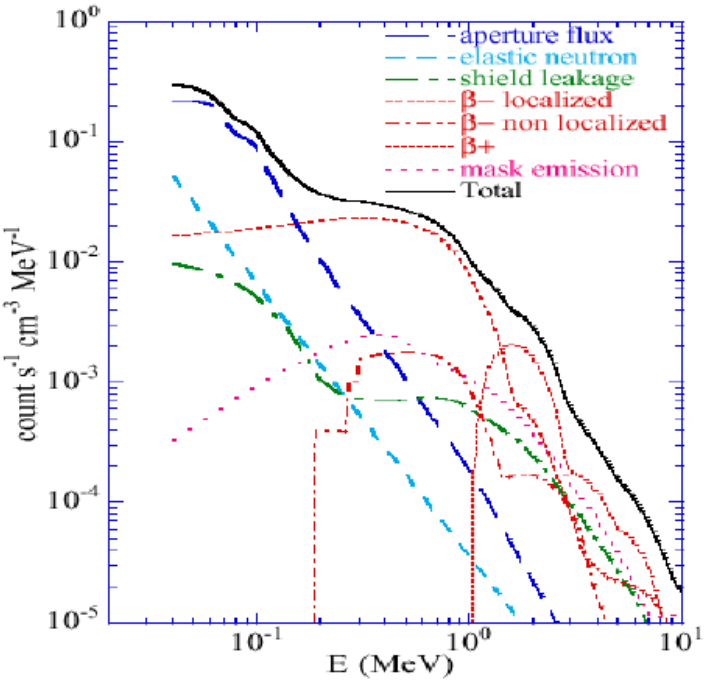


Figure 5: The continuum background components for SPI. The individual components are identified. The total background spectrum is indicated with the black line.

4.1.2 511 keV Background

The 511 keV background can be split into five components:

- the continuum background under the 511 keV line.
- 511 keV photons from passive material, due to β^+ decays of unstable nuclei in the materials. These are nuclei hit by protons and neutrons produced by interactions between cosmic ray particles and material from the detectors, shield or cryostat. The unstable nuclei decay through β^+ decay. The annihilation of the positron leads to the emission of two 511 keV photons in opposite directions. If one is absorbed by the detector and the other escapes, a 511 keV background event is produced.

- 511 keV photons, originating from interactions of cosmic rays with passive spacecraft materials, that are not rejected by the BGO shield (shield leakage).
- 511 keV photons originating from cosmic ray interactions with the mask material. The main source is pair creation by cosmic ray proton interactions with W nuclei. This component is significantly reduced with the Plastic Scintillator.
- BGO shield blocking time component: 511 keV photons produced by β^+ decays in the BGO shield when the ACS electronics is blocked by a large energy deposit and the veto is on.

All these components were reproduced with a Monte Carlo method. The resulting line strengths for the 511 keV line with and without the PSAC are given in Table 2.

component	with PSAC	without PSAC
continuum	4.9×10^{-5}	5.4×10^{-5}
passive material	2.5×10^{-4}	2.5×10^{-4}
shield leakage	8.6×10^{-6}	8.6×10^{-6}
mask	1.2×10^{-5}	2.1×10^{-4}
BGO shield blocking time	1.4×10^{-6}	1.4×10^{-6}
Total	3.2×10^{-4}	5.2×10^{-4}

Table 2: 511 keV background line strength with and without PSAC. Calculations are for 5 cm BGO shield, 80 keV shield threshold, without PSD. Fluxes in cts cm^{-3} s^{-1} .

4.1.3 Background Gamma-Ray Lines

Background gamma-ray lines are emitted in passive materials close to the detectors and in the detector material itself. Primary and secondary cosmic ray particles (protons and neutrons) induce excited nuclei in nuclear reactions with nuclei of the passive material. The prompt or delayed (radioactive) de-excitation of these nuclei leads to gamma-ray lines which can be detected by the germanium detectors. Calculations show that lines originating in the mask should not pose a problem for SPI.

4.2 Instrumental Characterization and Calibration

The SPI instrument has been fully tested and calibrated on ground before the launch. Soon after the launch the observations of the Crab nebula and Cygnus X-1 were done for further calibrations.

The detector gains, thresholds and resolution versus energy are determined from normal event data and ACS off spectra (for consistency checks) in the routine monitoring task of ISDC. Finally, after every detector annealing a thorough check is done of the instrument imaging and spectroscopic response, since these may change as a result of the annealing process.

4.3 Measured Performance

4.3.1 Imaging Resolution

The design of the instrument is such that the angular resolution for (isolated) point sources is about 2.5 °(FWHM). The location of point sources can be done with an accuracy much better than this, however, depending on the strength of the source. As explained above (Section 3.10), dithering is required, for imaging more complex regions.

4.3.2 Spectral Resolution

The spectral resolution has been measured in the laboratory with detectors that are representative of the flight units, and afterwards with flight model detectors and pre-amplifiers. An example spectrum obtained in this way is shown in Figure 6. The measured energy resolution as a function of energy for an individual detector is given in Figure 7.

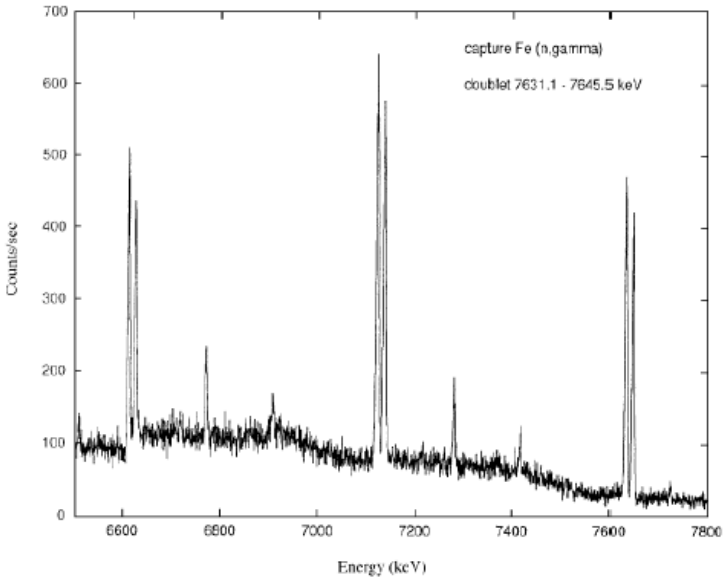


Figure 6: Example spectrum taken with laboratory detector units, representative of the flight units.

4.3.3 Sensitivity

The continuum and line sensitivities of the SPI instrument are given in Figure 8 and Figure 9. The continuum sensitivities are for $\Delta E = E/2$, and are calculated from the narrow line sensitivity by dividing those by $\sqrt{R \cdot \Delta E}$ where R is the instrument resolution for lines. The line sensitivities are fluxes in photons cm $^{-2}$ s $^{-1}$, the continuum sensitivities are fluxes in photons cm $^{-2}$ s $^{-1}$ keV $^{-1}$. The line sensitivities are for narrow lines. The 511 keV sensitivity is worse than the surrounding continuum due to the strong 511 keV background line originating in the instrument.

4.3.4 Dithering Sensitivity Degradation

The instrumental sensitivities given in Figures 8,9 are for a source on axis, and do not take into account dithering. As stated earlier, observations with SPI should not be done in staring mode, since this makes the identification and removal of the background impossible. Dithering on the other hand has the disadvantage that the source is not observed for the full integration time in the center of the fully coded field of view. The SPI response falls off towards the edge of the field of view, and therefore dithering will degrade the sensitivity of the instrument somewhat. The hexagonal dither (a central pointing with six surrounding pointings in hexagonal pattern, all 2°apart) only samples the central part of the SPI fully coded field-of-view. Therefore no reduction in the sensitivity is noticeable. The 5 by 5 dither however samples closer to the edge of the fully coded field-of-view. In this case the sensitivity is degraded by a factor 0.84.

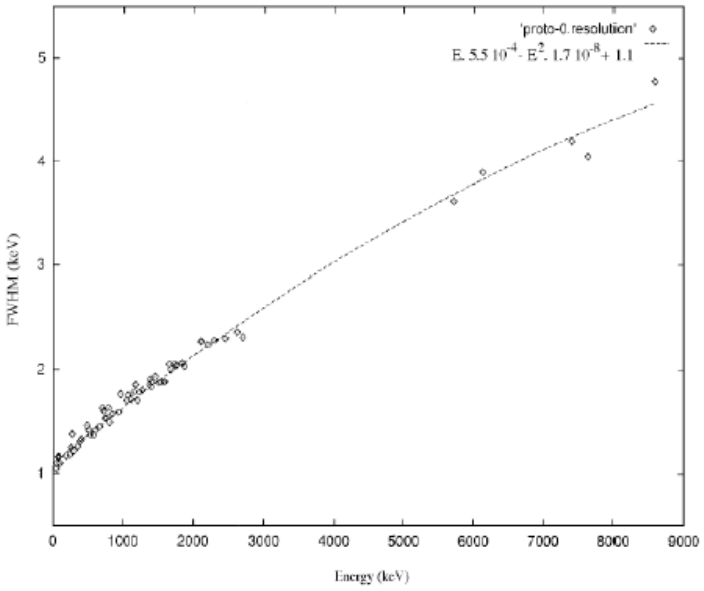


Figure 7: The measured energy resolution of an individual SPI detector. This was measured using laboratory detectors. The resolution of the full instrument with all 19 detectors is slightly lower than this.

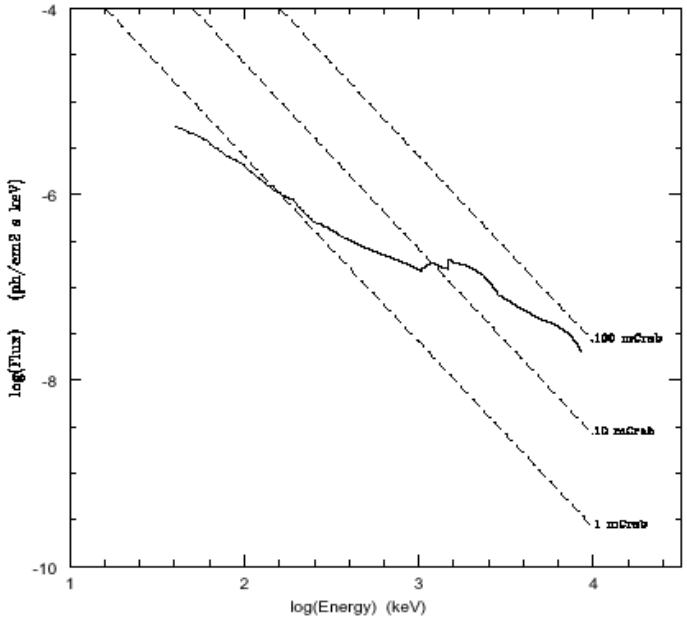


Figure 8: The continuum sensitivity of the SPI instrument for a 3 sigma detection in 10^6 seconds, on axis. The fluxes are for $E/\Delta E = 2$. The dashed lines indicate extrapolations from the X-rays using an powerlaw with photon index -2 for 1, 10 and 100 mCrab.

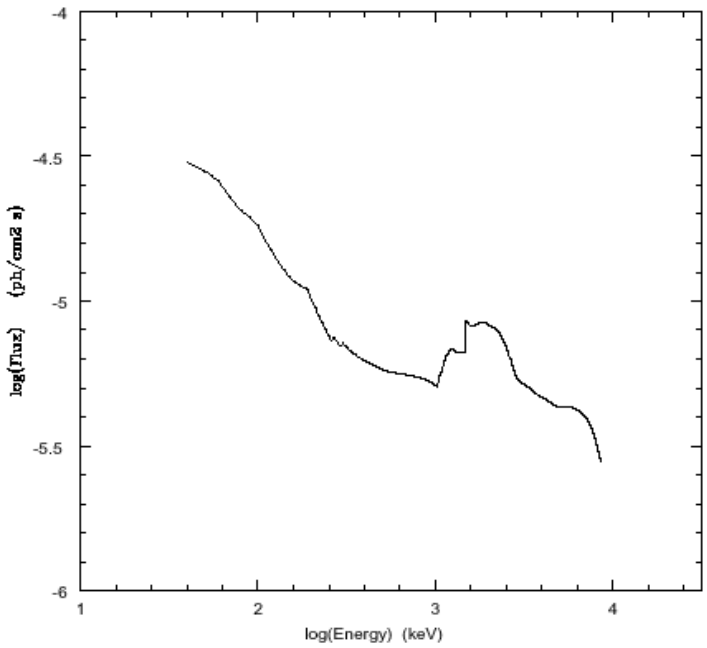


Figure 9: The narrow line sensitivity of the SPI instrument for a 3 sigma detection in 10^6 seconds. Note that the 511 keV line is not shown in this figure.

4.3.5 Detection of off-axis-sources

The wide field-of-view of SPI allows the detection of off-axis sources. However it also means that off-axis sources will create a shadowgram on the detector that increases the background photons for the prime target. To remove this background a proper mapping of the source and the surroundings is necessary. This is the main reason why hexagonal dithering should only be used for isolated point sources, where no significant contribution is expected from other sources within about 20°.

4.3.6 Imaging Capabilities

A thorough presentation of the SPI imaging capabilities can be found in the paper "The INTEGRAL spectrometer SPI: performance of point-source data analysis" published in MNRAS [4].

4.3.7 Timing Capabilities

In the SPI standard operation mode, each photon data set includes timing information given by a 100μs clock signal. This clock is synchronized to the on board clock, and thus to the UTC. The timing error budget for SPI is derived from

- the accuracy of the on-board clock and the synchronization,
- the conversion between on-board time and UTC,
- the conversion between UTC arrival time at the spacecraft and the arrival time at the solar system barycentre when appropriate.

The resulting SPI timing accuracy calculated in this way is 129μs, 3σ accuracy, and a 90% confidence accuracy of 94μs.

Part II

Data Analysis

5 Overview

The first task when starting a SPI data analysis is to identify and collect the input data. There are two main cases, (1) guest observers receiving their private data, and (2) users of the public data, browsing and retrieving data from the *INTEGRAL* archive. Basically, a list of pointings (see Glossary) have to be collected into an ASCII file. This ASCII file can then be fed into og_create to produce what is called an observation group (OG). The OG includes links to all input (and later output) data, and all software read/write data through this group.

Once you have created an OG, an image reconstruction or a spectral extraction can be achieved by running a single script "spi_science_analysis". This script analyzes the data going through a number of different steps briefly described in Table 3. These steps correspond to so-called "Levels" identified with 3-5 capital letter acronyms. Their meaning is common to all *INTEGRAL* instruments. A very important difference with the other instruments is however that (almost) always the complete set of pointings are analyzed together simultaneously. Apart for GRB analysis, single pointings are never analyzed independently.

Table 3: Overview of the successive steps of the spi_science_analysis script

Level	Program	Short description of the step
CAT_I	cat_extract	Extracts from the INTEGRAL reference catalogue information (position and flux) about known sources lying in the field-of-view.
POIN	spi_obs_point	Extracts the start/stop times and the SPI attitude for all pointings included in the observation, and stores the results in the *pointing.fits* file. On request, this program can access a list of bad pointings (usually of order of 1-2% of the total) and exclude them for the current analysis
BIN_I	spibounds	Defines the energy bins creating the *energy_boundaries.fits* file.
	spi_obs_hist	On-the-fly energy correction, event binning, and derivation of the dead times. Outputs three files (1) *gti.fits* containing "always good" GTI with ontimes, (2) *dead_time.fits* containing dead time ratios and livetimes, (3) *evts_det_spec.fits* containing the binned events formatted as spectra.
	spi_phase_hist	(Optional) event binning in phase intervals computed from provided ephemeris and taking into account possible orbital motion. Outputs one file per phase bin named *evts_det_spec_i.fits* where i is the phase bin number.
BKG_I	spi_obs_back	Computes background models from different indicators. Two main recommended options are (1) deriving the background from pre-defined *flat-field* files, and (2) using the Germanium saturated events as a background tracer. Outputs are stored into *back_model.fits*.
IMA, or SPE, or LCR	spiros	Carries out image reconstruction (IMA), spectral extraction (SPE) or timing analysis (LCR) depending on the chosen "mode". This is the most difficult and critical step. If the phase binning as been selected, all phase bins are processed independently in turn.
	spirmf	Invoked in spectral extraction only, creates the appropriate RMF response matrix required for XSPEC spectral fitting.

Since the release of OSA 8, another script is available for the spectral extraction and the contemporary fit of diffuse emission: spimodfit_analysis. This script performs all the steps of spi_science_analysis before running spiros. Instead of this program, the script calls the program *spimodfit* which allows for a different approach in the simultaneous fit of background, point sources, and diffuse emission using the SPi data. This will be described in Sect. 6.7 of the cookbook and with more detail in Sect. 8.7.

For analysis of Gamma-ray bursts a dedicated script spi_grb_analysis is available. This script is documented on-line at

http://isdc.unige.ch/index.cgi?Support+spi

6 Cookbook

We assume that you have already successfully installed the ISDC Off-line Scientific Analysis (OSA) Software (The directory in which OSA is installed is referred later as the ISDC_ENV directory). If it is not the case, look at the "Installation Guide for the *INTEGRAL* Data Analysis System" [2] for detailed help.

6.1 Setting up the analysis environment

6.1.1 Input data organization

The commands below apply to the `csh` family of shells (i.e `csh` and `tcsh`) and should be adapted for other families of shells[1]. In order to set up a proper environment, you first have to create an analysis directory (e.g `spi_data_rep`) and "cd" into it:

```
mkdir spi_data_rep
cd spi_data_rep
setenv REP_BASE_PROD  $PWD
```

This working directory will be referred to as the "REP_BASE_PROD" directory in the following. All the data required in your analysis should then be available from this "top" directory, and they should be organized as follow

scw/ : data produced by the instruments (e.g., event tables) cut and stored by ScWs,

aux/ : auxiliary data provided by the ground segments (e.g., time correlations),

cat/ : ISDC reference catalogue,

ic/ : Instrument Characteristics (IC), such as calibration data and instrument responses,

idx/ : set of indices used by the software to select appropriate IC data,

and all data resulting from your analysis will reside in an **obs** branch (in that same directory), which will be created by the first analysis step (og_create, see below). Part of the above input data may already be available on your system[2]. You can either copy the data to the relevant directory, or better, create soft links as follow

[1] If the setenv command fails with a message like:"setenv: command not found" or "setenv: not found", then you are probably using the sh family. In that case, please replace the command "setenv my_variable my_value" by the following command sequence "my_variable=my_value ; export my_variable". In the same manner, replace the command "source my_script" by the following command ". my_script" (the "." is not a typo!)

[2] The Instrument Characteristics files (OSA_IC package) and the Reference Catalogue (OSA_CAT package) are part of the OSA software distribution. They should be installed following the "Installation Guide for the *INTEGRAL* Data Analysis System" [2].

```
ln -s directory_of_ic_files_installation__/ic
ln -s directory_of_ic_files_installation__/idx
ln -s directory_of_cat_installation__/cat
ln -s directory_of_local_archive__/scw
ln -s directory_of_local_archive__/aux
```

The different "directory_of_..." can be all the same location. For example, if you are logged on the ISDC network and have a direct access to the revision 2 ISDC archive, these links become

```
ln -sf /isdc/arc/rev_2/ic
ln -sf /isdc/arc/rev_2/idx
ln -sf /isdc/arc/rev_2/cat
ln -sf /isdc/arc/rev_2/scw
ln -sf /isdc/arc/rev_2/aux
```

Alternatively, if you do not have any of the above data on your local system, or if you do not have a local archive with the scw/ and the aux/ branch available, follow the next section instructions to download data from the ISDC WWW site. If you do have already the data available on your system, you can skip next section (6.1.2) (you need however to build a list of "DOLs" for your analysis that you can either produced with the archive browser or "by hand" using shell command or/and an editor).

6.1.2 Downloading data from the ISDC archive

The below example is based on observations of the Crab from Revolution 102.

To retrieve the required analysis data from the archive, go to the following URL:

http://isdc.unige.ch/index.cgi?Data+browse

You reach the W3Browse web page which will allow you to build a list of Science Windows (SCWs) needed to create your observation group for OSA.

- Type the name of the object (Crab) in the "Object Name Or Coordinates:" field
- Click on the "More Options" button at the top or at the bottom of the web page
- De-select the "All" check box at the top of the Catalog table, and select the "SCW - Science Window Data" one
- Change the "Search Radius" from "Default" to "10 degrees" for our case.
- Press the "Specify Additional Parameters" button at the bottom of the web page
- De-select the "View All" check box (press twice on it) at the top of the Query table
- Select "scw_id" and put the value "0102%" (without the quotes) to specify all ScWs from Revolution 102
- Select "scw_type" and put the value "pointing" (without the quotes), or simply "po*" to get only pointings
- Select the "scw_id" for the "Sort" option (roundish button next to scw_id)
- Press the "Start Search" button at the bottom of the web page

At this point, you should be at the Query Results page with all the ScWs available for revolution 102. Sort the "Scw_id" column by clicking on the left arrow below the column Name

You can then select the ScWs we are interested in, i.e.,

```
010200200010
010200210010
010200220010
010200230010
010200240010
010200250010
010200260010
010200270010
010200280010
010200290010
```

Press the "**Save SCW list** for the creation of Observation Groups" button at the bottom of that table and save the file with the name "spi.lst" in your analysis current directory (i.e., $REP_BASE_PROD as explain above). The file "spi.lst" will be used later as an argument for the og_create program. In this file you find the lines

```
scw/0102/010200200010.001/swg.fits[1]
scw/0102/010200210010.001/swg.fits[1]
scw/0102/010200220010.001/swg.fits[1]
scw/0102/010200230010.001/swg.fits[1]
scw/0102/010200240010.001/swg.fits[1]
scw/0102/010200250010.001/swg.fits[1]
scw/0102/010200260010.001/swg.fits[1]
scw/0102/010200270010.001/swg.fits[1]
scw/0102/010200280010.001/swg.fits[1]
scw/0102/010200290010.001/swg.fits[1]
```

which is the list of DOLs (Data Object Locators) of the Science Window Groups you want to analyze. The DOLs includes the path and name of the FITS file, and the location of the SWG grouping table inside the file (remember that a FITS file can include more than one table, or image, arranged in a number of extensions). For example, the last DOL in the above list indicates a SWG located in the first extension of the FITS file "swg.fits"[3] lying in the directory "scw/0102/010200290010.001/".

You can download the data pressing the "**Request** data products for selected rows" button. In the "Public Data Distribution Form", provide your e-mail address and press the "Submit Request" button. You will be e-mailed the required script to get your data and the instructions for the settings of the IC files and the reference catalogue.

6.1.3 Setting the software environment

Before you run any OSA software, you must also set your software environment correctly. If not already set by default by your system administrator, you should set some environment variables, typing

```
setenv ISDC_ENV directory_of_OSA_sw_installation
setenv ISDC_REF_CAT "$REP_BASE_PROD/cat/hec/gnrl_refr_cat_0031.fits\[1]"
source $ISDC_ENV/bin/isdc_init_env.csh
```

[3]Note that the naming scheme is different for revision 1 and revision 2 data. For the revision 1 data, the name of the prepared Science Window Group is `swg_prp.fits` instead of `swg.fits`.

in order to:

- set ISDC_ENV to the location where OSA is installed,
- set ISDC_REF_CAT to the DOL of the ISDC Reference Catalog
- run the OSA set-up script (isdc_init_env.csh) which initializes further environment variables relative to ISDC_ENV.

The SPI pipeline creates a log file with all relevant information automatically each time it is executed. If you, however, want to log not only the output from the pipeline, but also from all your other OSA activities, set the environment COMMONLOGFILE:

```
setenv COMMONLOGFILE +name.log
```

where name.log is the name of the log file. You can use any name you want, but do not forget the + in front, otherwise, no information will be written to the screen any more.

As your level of expertise with the software increases, you may wish not to have the Graphical User Interface (GUI) pop up when you launch your analysis. In this case, the variable COMMONSCRIPT must be defined:

```
setenv COMMONSCRIPT 1
```

To revert to having the GUI, unset the variable:

```
unsetenv COMMONSCRIPT
```

6.2 Observation group (OG) creation

The first step of the analysis is to create the Observation Group (for more information on *og_create*, see [1]). The easiest is to launch the first program from the analysis top-level directory (i.e., $REP_BASE_PROD) so that the parameter "baseDir" can be set to the current directory.

```
og_create idxSwg=spi.lst ogid=spi_analysis baseDir="./" instrument=SPI
```

"spi.lst" is the file containing the list of pointing DOLs created as explained in above Sect. 6.1.2. Note that if this file is located in another directory, the full path should be specified together with its name."spi_analysis" is a user chosen parameter used to name the subdirectory where all analysis data will be written. "instrument" should never be changed for SPI data analysis. The result is an observation group named "og_spi.fits" located in "./obs/spi_analysis/". This later location is where the following of your analysis should be executed, and where all further analysis results will be written.

Depending on the number of science windows, the speed of your machine, and your data server, this can take from a few seconds for just one science window to more than an hour for hundreds of science windows (and of the order of a few seconds for this example).

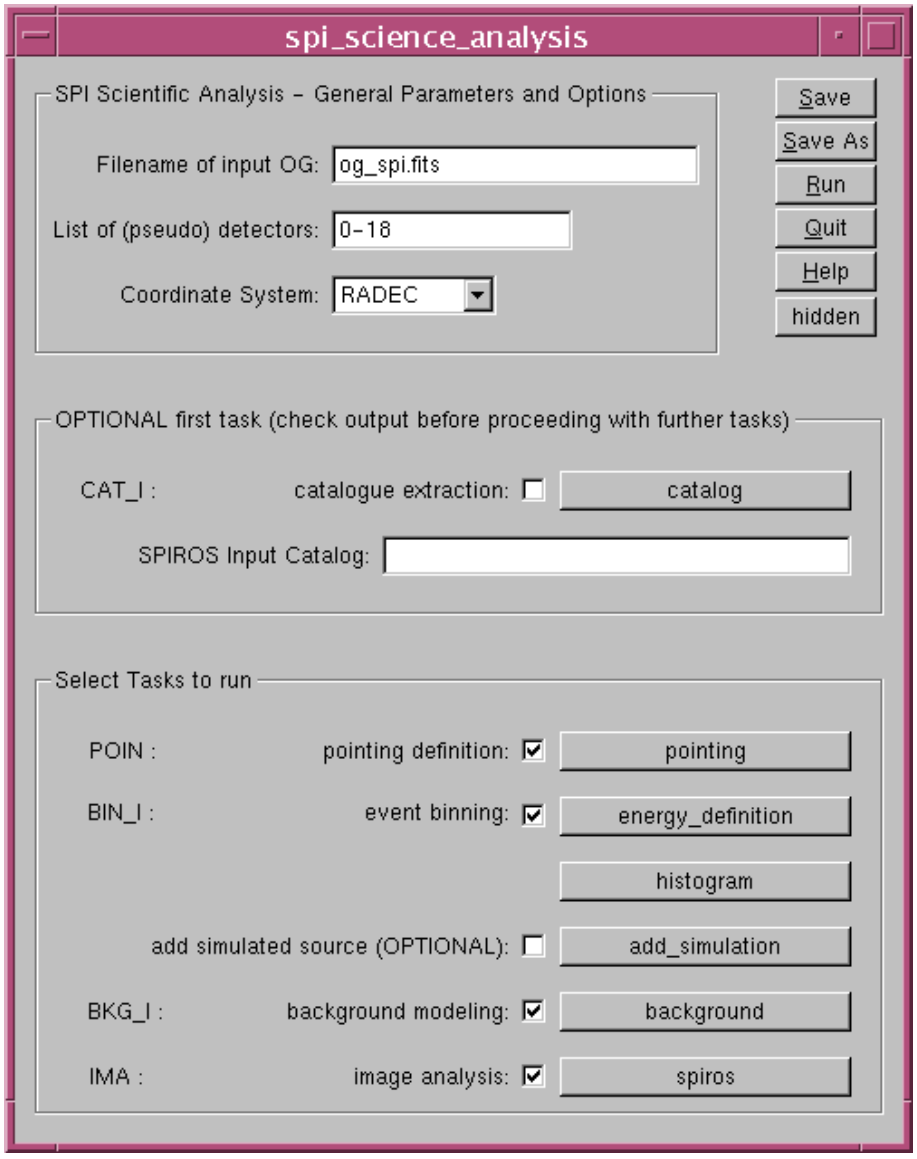
6.3 Image reconstruction without input catalogue

When *og_create* has finished without any errors, move from the top-level directory ($REP_BASE_PROD) into the analysis directory ("./obs/spi_analysis/") and launch the *spi_science_analysis* script

```
cd obs/spi_analysis
spi_science_analysis
```

After a few seconds the GUI will appear, provided that the DISPLAY environment variable is set properly, otherwise you will be prompted for the individual parameters on the command-line. In general, but especially for learning SPI data analysis, we **strongly** recommend using the GUI.

Figure 10: spi_science_analysis main GUI



The top section holds the most basic parameters of the pipeline

1. the list of detectors which are to be used in the analysis
2. the coordinate system

Do not change the default values for now. The middle section concerns the catalogue extraction/specification part. For most bright Gamma-ray sources, the available catalogue source posi-

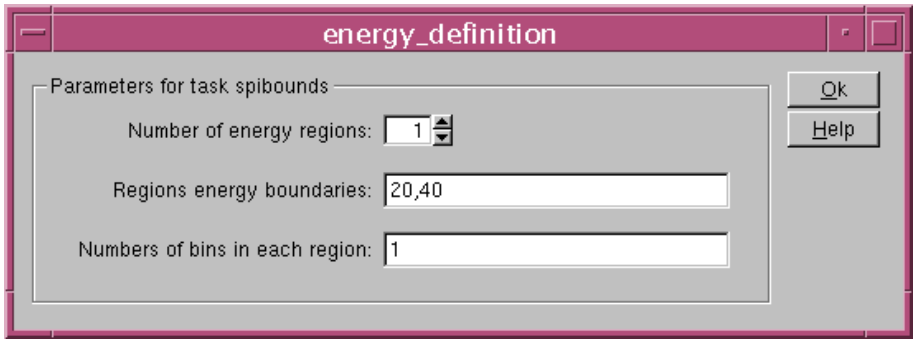
tions[4] are more accurate than the positions which can be derived from the SPI data (due to the limited spatial resolution). The best results are therefore obtained by fixing the known source positions in the image reconstruction process. However, as some sources are highly variable and the catalogue fluxes not always reliable, it is sometimes difficult to predict which sources will be detected in a given observation. In order to avoid specifying wrong positions (where no significant sources can be detected), it is a good idea to first explore your data set and see which sources can be detected without using any prior information. De select the "catalogue extraction" and leave the "SPIROS input catalogue" entry empty (we explain how to use this step in next Sect. 6.4.)

Select the pointing definition, event binning, background modeling, and image analysis tasks, and click on the corresponding buttons to specify task specific parameters.

1. For the pointing step, you can use a pre-defined list of bad pointings to automatically exclude from your analysis, pointings with known problems such as high background level or out-of-nominal instrument characteristics. This option is selected by default and we strongly recommend not to un-select it (or only for testing purposes).

2. In the "energy definition" (Fig. 11), you have three parameters. In the first field, you can define a number of energy regions. In the second one, you specify the actual limits in keV of these regions, i.e., 20,40,60 is for two regions: 20–40keV and 40–60keV. In the third field, you provide the number of bins for each energy region (positive numbers provide linear scaling, negative numbers logarithmic scaling). This allows for a more elaborate definition of the binning than a simple logarithmic or linear binning over the whole range. If you want gaps (i.e., 20–40keV and 80–100keV) in your binning, you simply set the number of bins in the region in-between to 0.

Figure 11: Energy bin definition GUI



   For our example, we use a 20–40keV energy region, with one bin.

3. The histogram option allows to run phase-resolved analyses, or alternatively to add a simulated source in the mono-energetic analysis. For this first example, do not change the default values and leave both options un-selected. They will be described in dedicated sections.

4. The background can be modeled using several options:

   (a) flat fields
   (b) templates
   (c) tracers

   In this example, we use the default flat field option, which is recommended by the SPI team. Pre-defined flat-field spectra are available in the `ic/spi/cal/` directory of the `REP_BASE_PROD`. They were derived by binning in 0.5 kev bins all events from empty-field observation revolutions, i.e.:

[4] either the position of a known counterpart from other-wavelength catalogues or a position derived from other Gamma-ray observations with higher spatial resolutions, such as the IBIS ones.

Figure 12: Main background GUI

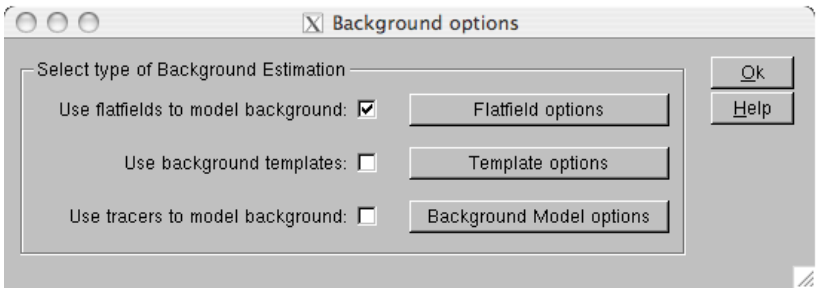
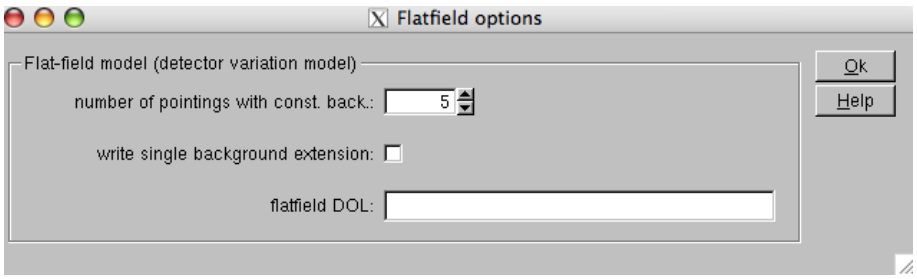


Figure 13: Options for the flat field background



- `spi_flat_grp_0012.fits`, revolution 72, to be used for all revolutions up to 140, i.e., with all 19 detector live,
- `spi_flat_grp_0013.fits`, revolution 158, to be used for revolutions 142 to 214, i.e., with 18 live detector (excluding number 2),
- `spi_flat_grp_0014.fits`, revolution 220, to be used for revolution 215 to 281, i.e., with 17 live detectors (excluding 2 and 17),
- `spi_flat_grp_0015.fits`, revolution 343-344, to be used for revolution 282 to 399, i.e., with 17 live detectors (excluding 2 and 17),
- `spi_flat_grp_0016.fits`, revolution 456-457, to be used for revolution 400 to 488, i.e., with 17 live detectors (excluding 2 and 17),
- `spi_flat_grp_0017.fits`, revolution 521-522, to be used for revolution 489 to 587, i.e., with 17 live detectors (excluding 2 and 17),
- `spi_flat_grp_0018.fits`, revolution 653-654, to be used for revolution 588 to 775, i.e., with 17 live detectors (excluding 2 and 17),
- `spi_flat_grp_0019.fits`, revolution 807-810, to be used for revolutions after 776, i.e., with 16 live detectors (excluding 2, 5, and 17).

These spectra are rebinned in this step to match the bin size of your analysis, and the program will select and use the appropriate flat-field spectrum automatically. However, the temporal proximity criterion used in the pipeline might not be the optimal one. In case of bad fitting results, the user is encouraged to choose another flat-field for the "flatfield DOL" in the panel of Fig. 13 (an appropriate path is required). With this option, the detector to detector ratios are kept fixed (from that of the empty field), but one background coefficient is derived through the imaging reconstruction process. In fact, one coefficient is derived for each time interval, and the length of the intervals is provided as a number of consecutive pointings by the user. To use the flat field model, select the flat field in the main background GUI and press the button for the flat fields options. Provide the number of pointings for which you assume the background to be constant. Use a value of 5 in this example und leave the "write single background extension" unselected. Note that in general, much longer time interval can be used in the range 20 to 100 pointings. The "write single background extension" must only be selected when using spimodfit instead of spiros (see the section on spimodfit, Sect. 6.7).

Another possibility to model the background accurately is to use background templates (the second option in the main background GUI). The background templates are provided by the SPI instrument team in Toulouse. These templates are derived from background tracers, but are fine tuned by the instrument specialists. Several templates exist based on various background tracers. A typical template is the GEDSAT which is based on the count rate of the events saturating the Germanium detector electronic (hence the name of the template). Other templates are based on the detector count rate, the non saturated events, and many more. The user is encouraged to try the various models to see which work best for his specific science and dataset. For typical point source analysis, a good starting point is the GEDSAT template. To use this model, un-select in the main background GUI the flat field option and select the background templates. Press the button for the template options and select the template to use.

Another way to model the background is the direct use of background tracers (the third option in the main background GUI). The GEDSAT model, which is also based on the count-rate of the events saturating the Germanium detector electronic is a good alternative to model the background. In this case, un-select the flat-field option and select the tracers option in the main background GUI. Press the Background model options button to set the options. Set the "Number of Models to use" to 1, and select the GEDSAT model. For more complicated setups you can use more components. The GUI offers the possibility to use up to four different model components simultaneously. In the image reconstruction process, we will assume that the real detector background varies in the same way as this model, which will be scaled to the actual data to determine the real background. GEDSAT is a good model for most point-source analyses (see Sect. 7 for more information about background modeling).

Figure 14: Options for the template background

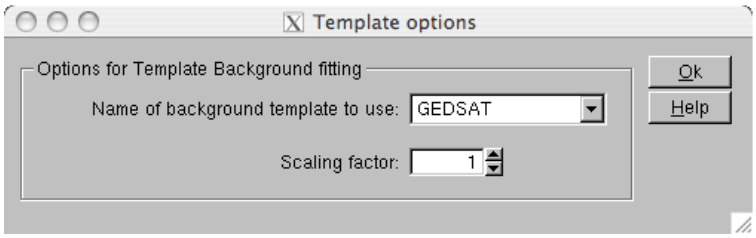
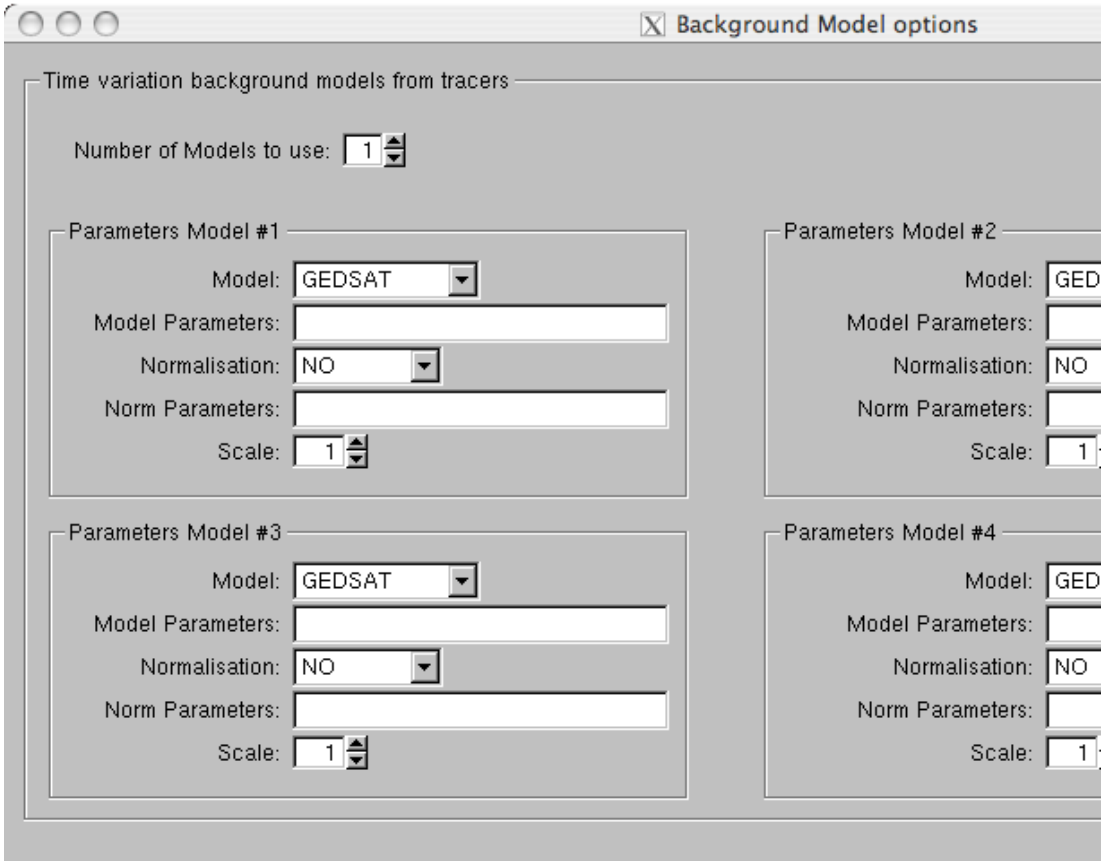
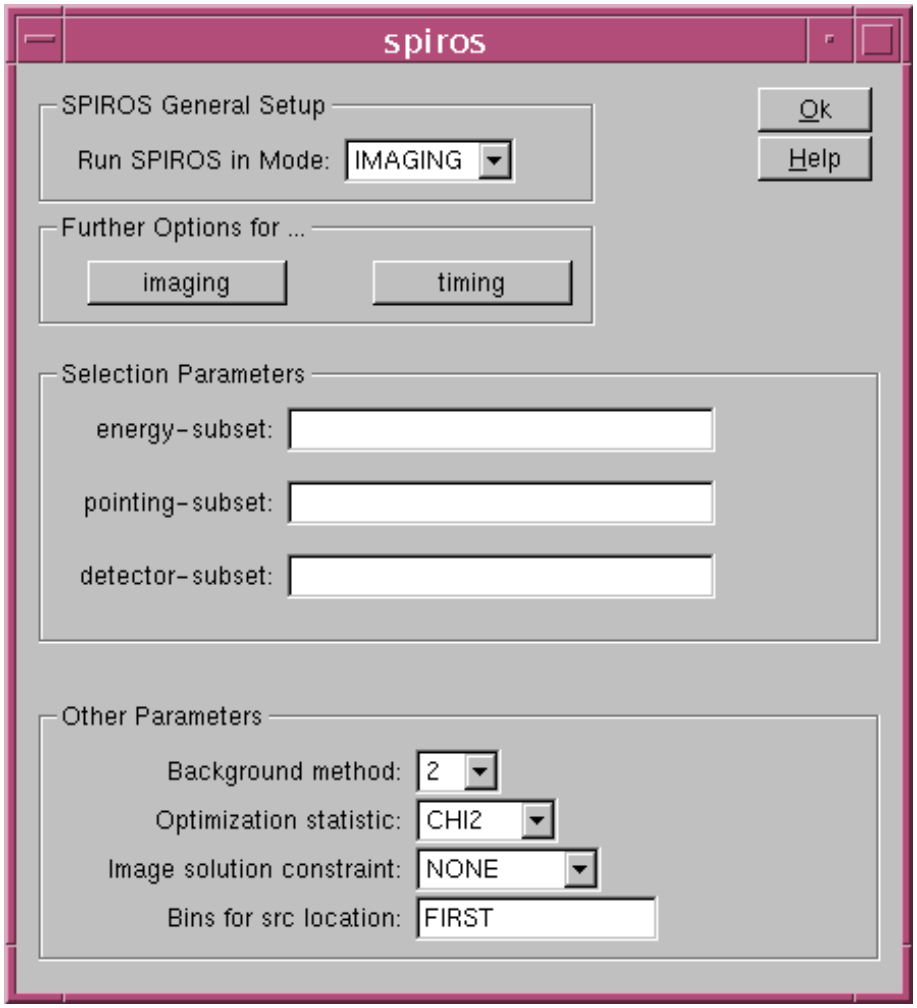


Figure 15: Options for modeling the background using tracers like GEDSAT



5. The last step of the SPI pipeline is the image deconvolution by spiros. Click on the spiros button to get the spiros GUI. Spiros performs image deconvolution in order to create images, spectra, or light curves. Since we want to create an image first, we set the spiros mode to IMAGING.

Figure 16: SPIROS main GUI



In the "Selection Parameters" fields of the spiros GUI, you can specify three subsets of energy bins, pointings, or detectors, respectively. If you leave the fields empty, spiros will use all data, otherwise only the specified energy bins, pointings, and detectors will be taken into account in the analysis. You can automatically ignore obviously bad data points but entering "AUTO" in the pointing subset. You can also use the AUTO filter in addition to specifying the pointings to use. For example, "AUTO,5-25,30-50" will use pointings 5-25 and 30-50 and apply the AUTO filter to these pointings. All other pointings will not be used.

The bottom section allows you to choose the "Background method", the "Optimization statistic", and the "Bins for src location".

For the background method, you must use method 3 if you selected the "flat-field" model. If you are using the GEDSAT model use method 2 (you can also use background method 5, also known as MCM, but be aware that this method may produce wrong results if you have bright variable sources in the field-of-view).

With spiros background method 3, one background scaling coefficient is derived for each time interval (specified as a number of pointings in the background model step) assuming that detector to detector background variations follow those from the flat-field model.

With spiros background method 2, one background scaling coefficient is derived for each detector in the image reconstruction process. In this way, we assume that the background variations in all detectors follow that of the background model (see Sect. 7.8 or the SPIROS User Manual for more explanations).

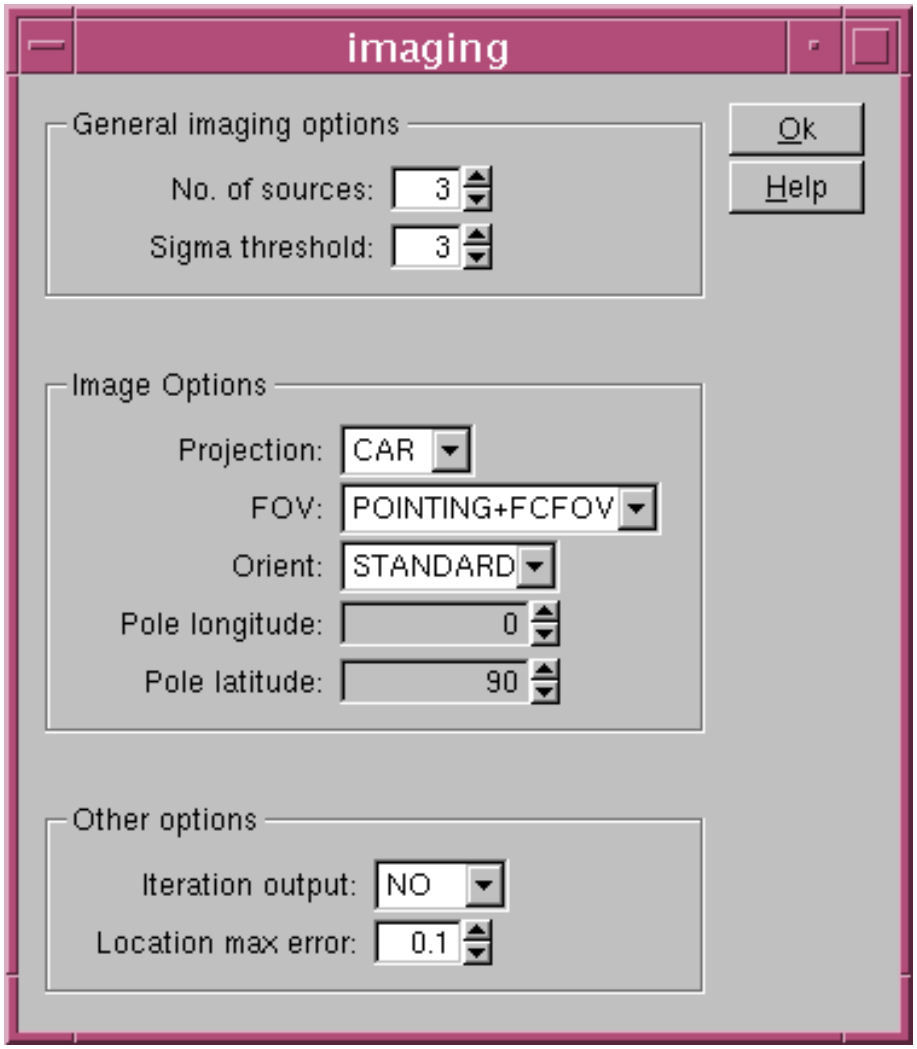
For the optimization statistic, you can choose between maximum likelihood or chi-square. The latter is faster and more robust but can be problematic if you have lots of energy bins with zero counts. For imaging in broad energy bands, chi-square is definitely better, while you might want to compare results obtained with chi-square and likelihood options when extracting spectra.

The bins for source location determines in which energy bins spiros locates the sources. The typical setting will be "FIRST" where spiros uses just the first energy bin. But you may also use ALL, where spiros will search for sources in each bin independently, or you may give a range of bins to be used.

For our example, we use background method 3, chi-square as optimization statistic, FIRST for the source location bins.

The options specific for imaging are available by clicking on the button "imaging" in the spiros GUI.

Figure 17: SPIROS imaging mode GUI



The top section allows you to select the maximum number of sources you want spiros to look for (in addition to those that might be provided through a catalogue, see Sect. 6.4) and a

lower sigma threshold. SPIROS stops its iterative source search process either when it has found the maximum number of sources or when the sigma threshold has been reached. In our case, we ask spiros to search for up to 3 sources with a detection significant larger than 3 sigma (enter 3 instead of the default 6).

We use the CAR (Cartesian) projection type for (almost) all cases. With the CAR projection the derived images have angular coordinates, with pixels regularly spaced by a constant angular step.

Since SPI is a coded mask instrument, you have to define the image field-of-view (FOV), i.e., the sky region that spiros should "reconstruct". An easy way to specify the image FOV is to use the POINTING option. The output image will then range from the minimum to the maximum of the central latitude/longitude values of all pointings included in your observation. Using "POINTING+FCFOV" or "POINTING+ZCFOV" will add half of the SPI fully-coded, or zero-coded FOV, on each side of the image defined by the pointing centers. "POINTING" or "POINTING+FCFOV" produces the nicest intensity images, but be aware that strong sources just outside your image FOV can completely bias your analysis (see Sect. 7) and is therefore only recommended when NO sources are close to FoV. If there sources close to the FoV, "POINTING+ZCFOV" is a good choice for most observations, however, the resulting intensity images need careful tuning in ds9.

When the setup is done, click twice "Ok" (once on "imaging" and once on "spiros"), then "Run" the pipeline. Depending on the number of science windows, the histogram building and image deconvolution can take quite some time.

Once the pipeline is finished (no errors should be apparent), you find in your directory amongst other output files the following three kinds of images, i.e., intensity, significance, and error

```
spiros_image_intensity_result.fits
spiros_image_sigma_result.fits
spiros_image_error_result.fits
```

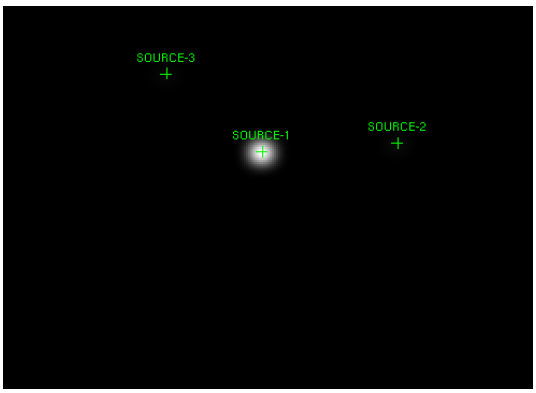
Before examining these images, convert the output catalogue produced by spiros into a format usable by ds9:

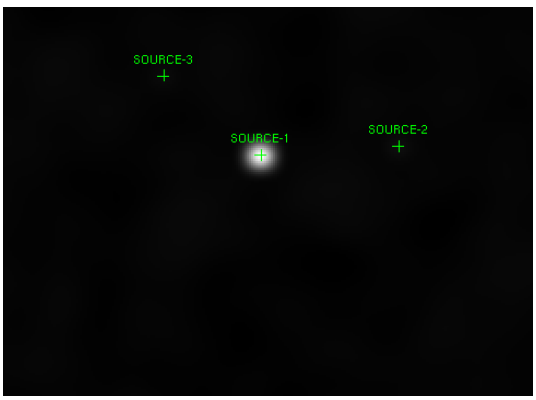
```
cat2ds9 "source_res.fits" source_res.reg

ds9 spiros_image_intensity_result.fits -region source_res.reg

ds9 spiros_image_sigma_result.fits -region source_res.reg
```

After adjusting the color tables etc, the resulting intensity and significance image should look like this:





There are three sources in these images. In addition to the obvious, strong Crab detection, spiros has found two spurious sources with very low significance. In most cases, spiros does find spurious sources with detection significance in the range 3 to 5 sigmas. Great care should be taken to make sure that sources with such detection significance are indeed real, and if you only want to see sources detected "beyond any doubt" raise the spiros lower sigma threshold to higher values (6 is usually a good choice).

The output file

```
source_res.fits
```

provides quantitative results formatted as a FITS table. Open this file with the ftool "fv" to look at the results. The resulting spiros Crab position is RA=83.617 DEC=22.019, while the catalogue position is RA=83.633 DEC=22.014. The difference is of the order of 1 arcmin. As expected for very high S/N sources, this value is much smaller than the SPI spatial resolution. The Crab flux in this 20-40 keV energy band is 0.175 ± 0.001 ph/sec/cm^2, corresponding to a detection significance of 159. You can also see that the detection significance of the spurious sources is 5.7 and 3.8.

All these results can also be viewed, perhaps more conveniently, in the end of the log file (called spi_sa_"UT date".log), below the following title

```
================================================================================
=                                                                              =
=        Summary of results from SPIROS-9.3.2 in IROS analysis mode            =
=                                                                              =
================================================================================
```

where you find the most important information about the analysis, such as information about the sources found at each iteration and about the resulting images.

6.3.1 Data screening

Near the end of the – spi_sa_"UT date".log – log file, you can find a table with chi square values per pointing looking as follow (with a slightly different formatting however):

```
Contributions to CHI2 parameter by pointing exposure.

 Ptg Rev Exp  ONTIME  CHI2,ML Expected Diff  Reduced Data
 -no -no -no  (secs)   value   value   /STD  CHI2,ML excl
 ------------------------------------------------------
  1 102  20  2155.2    21.5    18.0   0.58   1.19   0.00
  2 102  21  1932.7    19.6    18.0   0.27   1.09   0.00
  3 102  22  2200.8    49.4    18.0   5.24   2.75   0.00
  4 102  23  2198.8    40.0    18.0   3.67   2.22   0.00
  5 102  24  2196.8    19.2    18.0   0.20   1.07   0.00
  6 102  25  2143.8    24.9    18.0   1.15   1.38   0.00
  7 102  26  2198.8    30.9    18.0   2.16   1.72   0.00
  8 102  27  2196.8    36.9    18.0   3.15   2.05   0.00
  9 102  28  2194.8    22.7    18.0   0.79   1.26   0.00
 10 102  29  2189.8    16.6    18.0  -0.24   0.92   0.00


 Ptg Rev Exp  ONTIME  CHI2,ML Expected Diff  Reduced Data
 -no -no -no  (secs)   value   value   /STD  CHI2,ML excl
```

```
-----------------------------------------------------------
  1 102  20  2155.2    19.1    16.3   0.49   1.17   0.00
  2 102  21  1932.7    15.7    16.3  -0.11   0.96   0.00
  3 102  22  2200.8    43.5    16.3   4.76   2.67   0.00
  4 102  23  2198.8    31.5    16.3   2.66   1.93   0.00
  5 102  24  2196.8    19.3    16.3   0.53   1.19   0.00
  6 102  25  2143.8    18.5    16.3   0.38   1.13   0.00
  7 102  26  2198.8    18.3    16.3   0.36   1.12   0.00
  8 102  27  2196.8    19.8    16.3   0.61   1.21   0.00
  9 102  28  2194.8    30.9    16.3   2.55   1.89   0.00
 10 102  29  2189.8     9.7    16.3  -1.16   0.59   0.00
```

The most important information to consider is "Diff/STD", which gives the difference, expressed in term of standard deviation, between the CHI2 (or ML) resulting values and the expected ones. In our case, these residuals appear reasonable for all pointing with the exception of number 3 (with 5.24 sigma). When using your own data, you may find some pointings with very large residual values. DO NOT TRUST THE RESULTS if some of the involved pointings have residuals much larger than 3 standard deviations (although see Sect. 7 for possible exception to this rule). You must get rid of these residuals. The first thing to do is to use the AUTO filter of spiros. Launch spi_science_analysis again, de select all tasks except spiros and enter "AUTO" in the pointing subset under spiros options and run the pipeline again. The AUTO filter automatically removes all data on a per pointing and per detector base that have completely unreasonable values. For most datasets, this significantly improves the results, but not in our case. The next thing to do is to work on the background. You should use different background models to improve the modelling. We will use the MCM method in the next step: launch spi_science_analysis again, open the spiros options, and select background method 5 and run the pipeline. The results are now much better:

```
Ptg Rev Exp  ONTIME  CHI2,ML Expected Diff  Reduced Data
-no -no -no  (secs)   value   value   /STD  CHI2,ML excl
-----------------------------------------------------------
 1 102  20  2155.2    17.3    15.3   0.36   1.13   0.00
 2 102  21  1932.7    16.7    15.3   0.26   1.09   0.00
 3 102  22  2200.8    20.3    15.3   0.90   1.32   0.00
 4 102  23  2198.8    26.8    15.3   2.09   1.75   0.00
 5 102  24  2196.8    24.7    15.3   1.69   1.61   0.00
 6 102  25  2143.8    16.3    15.3   0.18   1.07   0.00
 7 102  26  2198.8    22.6    15.3   1.31   1.47   0.00
 8 102  27  2196.8    15.4    15.3   0.01   1.00   0.00
 9 102  28  2194.8    19.2    15.3   0.71   1.26   0.00
10 102  29  2189.8    11.3    15.3  -0.73   0.74   0.00
```

All residuals are now below three sigma and we do not need to further work on the extraction. However, in other datasets you may still have pointings with significant residuals. In this case you may have bright variable sources in the FoV. See Sect. 7 on how to deal with such cases.

Nevertheless you will encounter pointings with high residual values, which you have to remove from your analysis to obtain reliable results. To exclude such bad pointings from your analysis, note the pointing numbers of the bad pointings. Then launch spi_science_analysis again, de-select all tasks except spiros. Open the "Spiros options" GUI. Enter all **good** pointings in the **pointing subset** entry field. I.e., if pointing 7 and 8 of a total of 20 pointings are bad, enter 1-6,9-20 as pointing subset. Now rerun the analysis. You will find that this time the resulting images are much better. If you check the log file, you'll see that the chi square values of all pointings have improved significantly. If you still encounter some pointings with bad chi square values, simply repeat the procedure until all chi square values are reasonable. But beware: it is easy to obtain results with very small residuals by removing all bad pointings from the analysis. However, you must be very careful when removing pointings: all really bad pointings are removed automatically

by the pipeline during the pointing step and by the AUTO filter of spiros. Very often when you still encounter bad pointings, these are due to an insufficient sky model (i.e., you may have a strong variable source or a transient in the FoV)!

6.4 Image reconstruction with catalogue information

Once you have seen which sources are detected by spiros, you can obtain better results by fixing the source positions in the image reconstruction. For example, you may want to fix the position of the bright, well known sources to see whether the detection significance of a fainter source improves (as it should in case of a real detection).

First, you may want to save the previous results. The easiest way is to make a full copy of your current working directory:

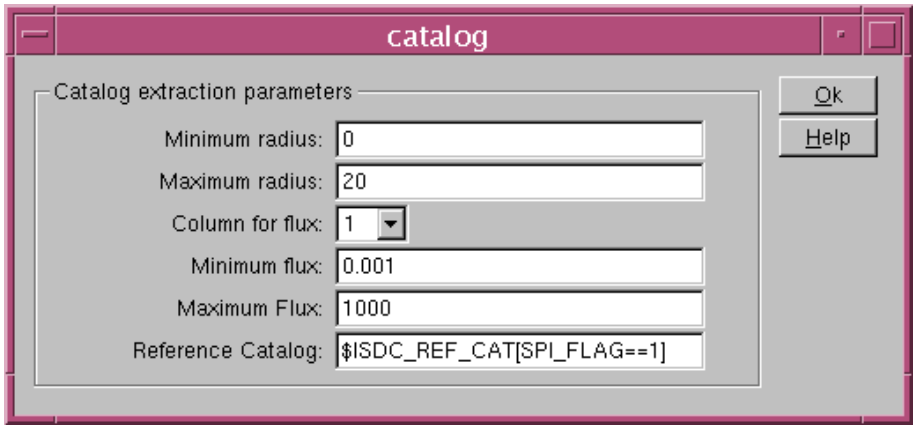
```
cd ..
cp -r spi_analysis spi_imaging_without_catalogue
cd spi_analysis
```

The source positions have to be provided to spiros through a source_cat.fits file. You can either use an existing file or create a new one with the catalogue extraction task (middle section of the main GUI). Here we explain how to produce a new source_cat.fits file including sources found in the ISDC reference catalogue in the observation field-of-view.

6.4.1 Creating a new source_cat.fits file

Launch spi_science_analysis and select the catalogue extraction. We strongly advise to stop the script after this step and to check the results before going again further into the analysis. This is done by de-selecting all tasks but catalogue extraction.

Figure 18: Catalogue extraction GUI



To change the catalogue extraction options, just click the catalog button and the catalog GUI will appear. Options are the maximum and minimum fluxes and you can provide an optional reference catalogue to use. If you want the resulting source list to contain only sources which have been detected with SPI before (for most observations this is a good idea), enter $ISDC_REF_CAT[SPI_FLAG==1] in the reference catalog field (provided that the environment variable ISDC_REF_CAT is set which should be the case). When you are done with your changes, click OK and run the pipeline.

When the pipeline has finished, analyze the contents of the catalog:

```
fv source_cat.fits
```

Review the catalogue carefully. If you don't like what you see (e.g., the sources you are interested in are missing), quit fv, restart *spi_science_analysis* and change the options for the catalogue extraction step (i.e., lowering the flux threshold if you're source is weaker than the threshold). For minor modifications or removing entries use fv to modify the catalogue according to your needs.

In our case, depending on the catalogue version used, one or two sources ("H 0614+091" and/or "X per") may be selected in addition to "Crab". We do not want to use these sources in the analysis. Using the "fv" menu, find the SEL_FLAG column (one before the last one) and enter SEL_FLAG = 0 for these sources. Source with a selection flag of zero will be ignored by spiros (alternatively you can also removed the un-wanted rows). Save the catalogue and quit fv.

See Sect. 7 for advices on how to use catalogue information with spiros.

6.4.2 Using a source_cat.fits file in the analysis

In order to use fixed source positions in an analysis, launch "spi_science_analysis", de-select the catalogue extraction step and provide the name of your catalogue file in the "SPIROS input catalogue" entry field. We advise to use the standard name "source_cat.fits" for input catalogues.

At this stage, the complete analysis can be run again (and previous results will be overwritten). However, the catalogues information is only used in spiros, and if there is no need to change other analysis parameters (such as the energy binning) only this last step can be repeated. Select "Image analysis", click on "spiros", and then on "imaging". You can then change the maximum "No. of sources" to 2 (instead of 3). This change is not really critical, but for a better comparison with the previous analysis we would like to restrict spiros to a 3 source analysis. Since the Crab source is now provided through the catalogue, we specify a maximum of 2 "additional" sources. Click twice on "Ok" to close the spiros GUIs, and "Run" on the main GUI. The script will execute spiros and stop.

The resulting images can be viewed with ds9 and the "source_res.fits" file with fv as explained before. The log file situation is now a bit more complicated. If you type

```
ls -rtl spi_sa*.log
```

you will notice three log files. In fact, the spi_science_analysis script creates a new log file, with a "spi_sa" + UT date + ".log"naming scheme, each time it is invoked. In our case, the first log file contains outputs from the previous analysis (without catalogue inputs), while the second file includes outputs from the catalogue extraction step. In the third file, you can find the outputs of the spiros "with catalogue inputs" run, below the usual heading

```
================================================================================
=                                                                              =
=         Summary of results from SPIROS-9.3.2 in IROS analysis mode           =
=                                                                              =
================================================================================
```

Looking in some details at this log file, you will notice that spiros provides the name of the catalogue that is used, and that it is explained that one source was provided through the catalogue while two other sources have been detected. The results (images, source_res.fits, log files) are almost identical to those of the previous case, however, you will notice that the Crab position (in source_res.fits as well as in the log file) is exactly the same as the catalogue one.

6.5 Spectral Extraction

Before starting, you may want to save your imaging analysis again, by copying the current directory

```
cd ..
cp -r spi_analysis  spi_imaging_with_catalogue
cd  spi_analysis
```

The SPIROS program can only extract spectra at fixed, known positions. Therefore, a source_cat.fits file must be provided in this case. We assume that you have created/checked a source_cat.fits as explained in last section. Launch "spi_science_analysis", de-select the catalogue extraction step, and provide the input catalogue name `source_cat.fits` in the "SPIROS input catalogue" field.

It is not necessary to run the pointing step again. However, the number of considered energy bins is typically much larger in spectral extractions than in imaging reconstructions. Select, the binning, background, and spiros steps. In the energy binning step, you can now select a binning suitable for a spectrum of a bright source with a broader energy range and a relatively high number of bins. For our example, set the number of regions to "1", a "20,1000" keV boundary, and "−100" as the number of bins. Using a negative bin number implies a logarithmic binning, with 100 bins from 20keV up to 1MeV in our case.

Open the spiros GUI and set spiros mode to spectra instead of imaging. Click "Ok" and "Run" the pipeline.

After spiros has finished successfully, the pipeline will automatically generate the appropriate response matrix and attached it to your spectra.

Now launch XSPEC to analyze your spectra. In the following example we fit a single power law from 30 to 1000keV. Here is a verbatim of the XSPEC session.

```
================================================================================

               Xspec 11.3.1    17:41:21 26-May-2005

 For documentation, notes, and fixes see  http://xspec.gsfc.nasa.gov/

 Plot device not set, use "cpd" to set it

 Type "help" or "?" for further information
XSPEC>data spectrum_Crab.fits
POISSERR keyword not found, assuming FALSE
 Net count rate (cts/s) for file   1  0.3903    +/-  3.3501E-03
   using response (RMF) file...       spectral_response.rmf.fits
   1 data set is in use
XSPEC>cpd /xs
XSPEC>setp en
XSPEC>plot ld
XSPEC>ign **-30.
 ignoring channels     1 -    11 in dataset     1
XSPEC>model power
  Model:  powerlaw<1>
Input parameter value, delta, min, bot, top, and max values for ...
                   1      0.01        -3        -2         9        10
1:powerlaw:PhoIndex>
                   1      0.01         0         0     1E+24     1E+24
2:powerlaw:norm>
```

```
  ------------------------------------------------------------------------
  ------------------------------------------------------------------------
  Model:  powerlaw<1>
  Model Fit Model Component  Parameter  Unit     Value
  par   par comp
    1    1    1   powerlaw   PhoIndex            1.00000     +/-   0.00000
    2    2    1   powerlaw   norm                1.00000     +/-   0.00000
  ------------------------------------------------------------------------
  ------------------------------------------------------------------------
 Chi-Squared =      4010090.     using    89 PHA bins.
 Reduced chi-squared =      46092.99     for     87 degrees of freedom
 Null hypothesis probability =  0.00
XSPEC>fit 50

<---snip--->

  ------------------------------------------------------------------------
  Model:  powerlaw<1>
  Model Fit Model Component  Parameter  Unit     Value
  par   par comp
    1    1    1   powerlaw   PhoIndex            2.16210     +/-  0.123967E-01
    2    2    1   powerlaw   norm                13.3916     +/-  0.626640
  ------------------------------------------------------------------------
  ------------------------------------------------------------------------
 Chi-Squared =      115.7470     using    89 PHA bins.
 Reduced chi-squared =      1.330425     for     87 degrees of freedom
 Null hypothesis probability = 2.137E-02
XSPEC>plot ld de

================================================================================
```

The final "plot ld de" produces the following spectral display.

Note that the remaining deviations are indeed really small, especially considering that we are using only 10 science windows in this example which is a very small number for SPI.

6.6 Light-curves

SPIROS can only produce light curves for sources provided through an input catalogue. The easiest starting point in this cook book is to copy the data set used for "imaging with a catalogue" for which we used a single energy bin from 20-40 keV. The command below are to save the results of spectral analysis and to start again with the imaging data set (Note that the below naming is somewhat arbitrary, working directly in a spi_timing directory could also be meaningful).

```
cd ..
mv spi_analysis spi_spectra
cp -r spi_imaging_with_catalogue spi_analysis
cd spi_analysis
```

Launch "spi_science_analysis" check that an input catalogue is provided and de-select all tasks except spiros. Open the "Spiros options" GUI and set the spiros mode to TIMING. Click on the timing button to get the timing GUI.

Set the mode to QUICKLOOK and the Timing Scale to 0. The Timing Scale is the time resolution of the resulting light curve in *days*. Setting it to 0 will give you one data point per science window

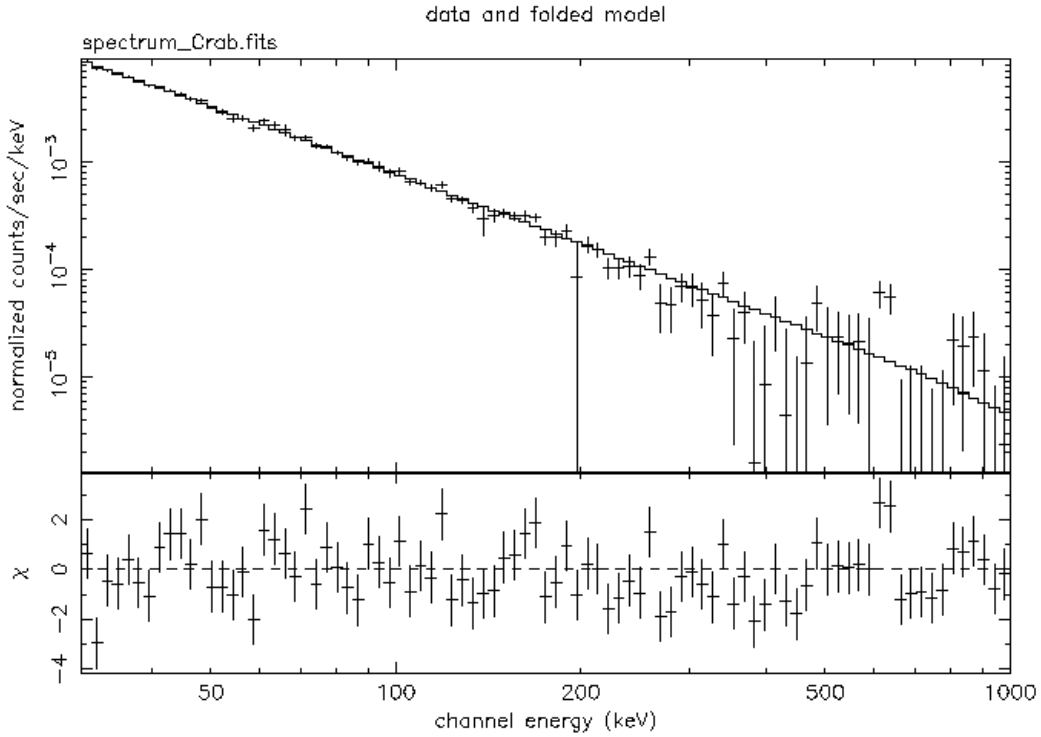
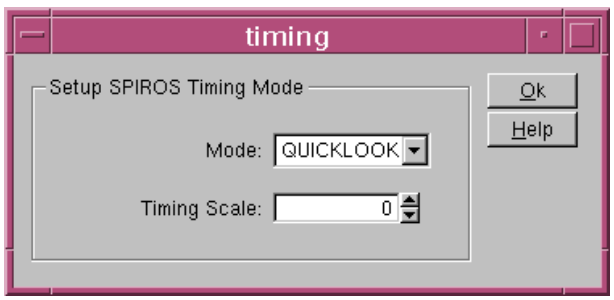


Figure 19: Crab spectrum from cookbook example

Figure 20: SPIROS timing mode GUI



– the highest time resolution possible. At the moment it is not possible to create light curves with a resolution smaller than one science window.

Click “Ok” twice and then “Run”.

The resulting light-curve is written to **spiros_timing_data.fits**. This file contains one extension for each source plus the background (first data extension). This light-curve can be displayed with e.g. “fv” to produce the output presented in Fig. 21.

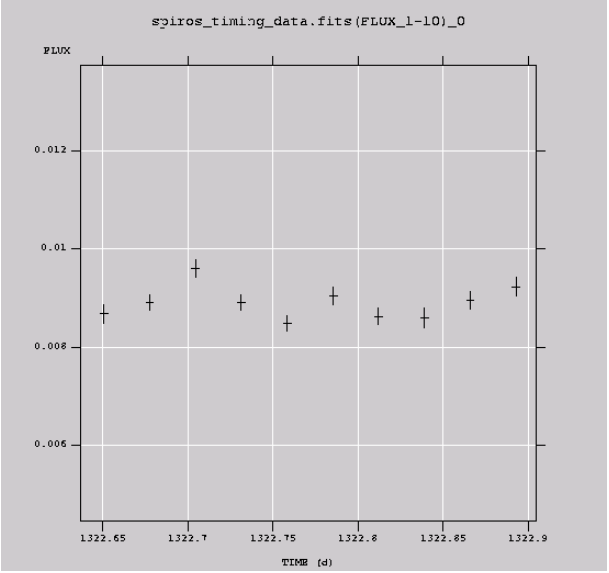


Figure 21: Crab light-curve from the cookbook example

6.7 Spectral Extraction using *spimodfit*

First, set-up the environment and create the observation group as detailed in the sections for `spi_science_analysis`. Then select a suitable catalogue for the observation you are analyzing. Using the cookbok example the name is `source_cat.fits`, you should leave this name. In our example we set the column FLAG to the value 1 for Crab.

Then start `spimodfit_analysis` and select the steps POINT BIN_I and FIT (Fig. 22). The user can skip the background step, since *spimodfit* handles it internally.

Then click on energy definition and select the number of channels you want in the final spectrum, we chose 22 logarithmic bins from 25 and 300 kev (Fig.23). Close this window and open the window on “spimodfit options” (Fig. 24) and choose correctly in the upper left frame the selected bins according to your binning choice. You can rebin at this stage with an integer number of bins. All the opther parameters can be left to the default values.

Several files are produced as output, one should check the fit quality on the program log `spi_sa_DATE.log`, where `DATE` is the start time of the analysis variable from run to run. For each energy range, there is the value of the χ^2_ν and of the C statistics at optimum. These should be around one, if a $\chi^2_\nu > 10$ is obtained, a warning is issued and the user should consider to vary the input catalog, the variability time scales, and/or the background settings.

For the user interested in spectral analysis of point sources the relevant output are the spectra named `spectrum_SRC.fits` and the response matrix `spectral_response.rmf.fits`, no ancillary effective area is necessary. In our case the Crab spectrum is (`spectrum_Crab.fits`) whose fit with a power law yields a spectral index 2.10 ± 0.03 and normalization at 1 keV 11.0 ± 0.7 (Fig. 26). The

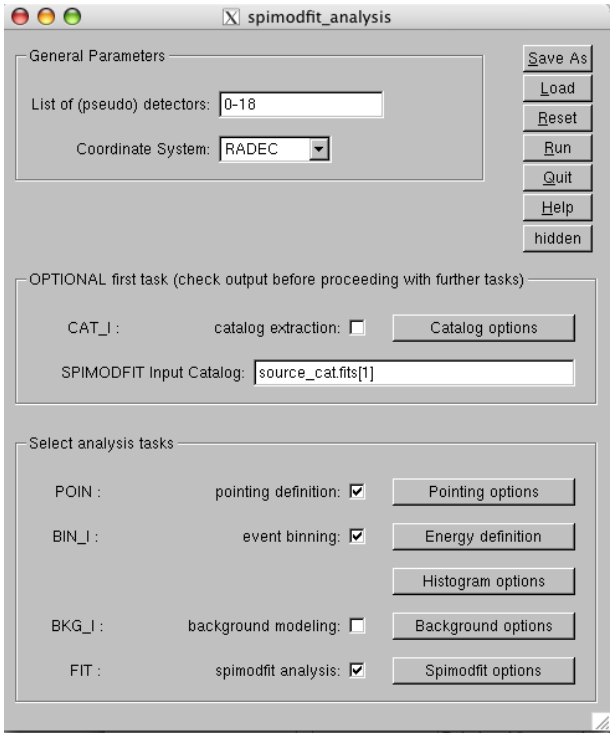


Figure 22: GUI of `spimodfit_analysis` with the default setting.

response matrix is computed by the script `spimodfit_rmfgen.csh`, which should be run manually in case the file `spectral_response.rmf.fits` is not present.

To run the fitting step another time the script automaticcaly deletes the previous output files with the command
`rm -f spimodfit_*`, therefore if the user wants to compare the results of subsequent runs, he/she should copy the relevant files elsewhere. At odds with the `spi_science_analysis` pipeline, the results of spimodfit are not attached to the `og_group`, therefore the deletion of these files does not produce any harm to the og_tree.

6.7.1 Further settings

The central frame allows the user to define a source variability time scale. This is applied to all sources in the local catalog and is a delicate feature since it increases the number of parameters that are fitted. The default is non variability, but different choices can be done for highly variable sources, especially in the crowded fields. The format of the variability defines nodes "IJD d n", where IJD is the `INTEGRAL` Julian day, and intervals "DD d i", where DD is the variability timescale in days. The variability should be applied only to energy bins with high enough S/N, typically at the lower energy range for SPI. It is not currently possible to separate within the pipeline different spectral regions, therefore subsequent runs of *spimodfit* should be done to allow spectral variability only in some bands and the results combined by the user, using his/her own tools.

On the upper right frame of the GUI (Fig.23), we provide the functionality for simultaneous fitting of diffuse emission and point sources, but we encourage the reader to consult the specific handbook of *spimodfit* to understand how this is handled. We suggest the general user not to exploit this functionality, unless he/she knows well the instrument and the program.

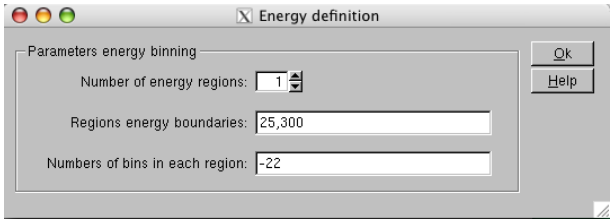


Figure 23: GUI of `spimodfit_analysis` energy definition, here 22 equally spaced logaritmic bins have been selected.

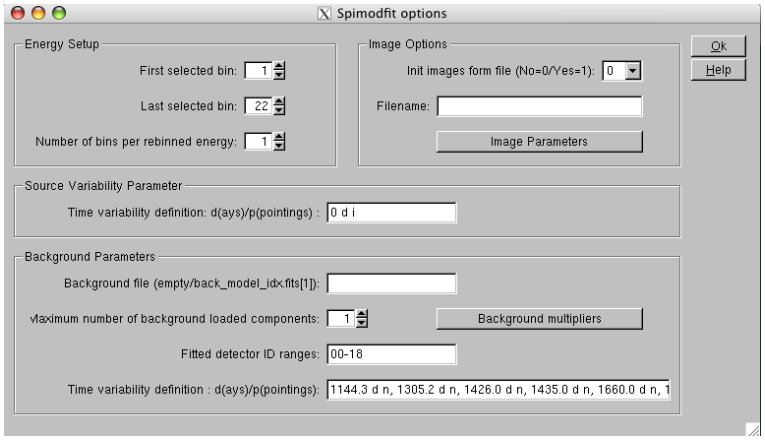

Spimodfit options
Energy Setup
First selected bin: 1
Last selected bin: 22
Number of bins per rebinned energy: 1
Image Options
Init images form file (No=0/Yes=1): 0
Filename:
Image Parameters
Ok
Help
Source Variability Parameter
Time variability definition: d(ays)/p(pointings) : 0 d i
Background Parameters
Background file (empty/back_model_idx.fits[1]):
Maximum number of background loaded components: 1
Background multipliers
Fitted detector ID ranges: 00-18
Time variability definition : d(ays)/p(pointings): 1144.3 d n, 1305.2 d n, 1426.0 d n, 1435.0 d n, 1660.0 d n, 1

Figure 24: GUI of `spimodfit_analysis` regarding *spimodfit* .

The bottom frame regards the background settings. The entry for the background file can be left empty to allow the internal handling, this is our suggested choice, or can assume the value `back_model_idx.fits[1]` to use the background computed in the background step. The following entry regards this case, so that the user can allow for multiple background components, this is an advanced feature that should be used with a deep knowledge of *spimodfit* . The entry with the fitted detector ID range allows the user to select different ranges of the detectors to fit the background, the choice 00–18 is suggested so that the background at this stage is handled as a constant. The entry time variability definition allows the user to define time variability nodes and intervals. The default values regard the SPI annealing times and the detector failures plus a variability on science window time scale (0.084 days).

The button on background multipliers opens a window (Fig 25) through which the user can set-up the background multipliers. This is a features that allows the program to compute a background pattern. The relevant parameters are the detector ranges and the Variability coefficients. The former is by default the list of all detectors so that each detector can be treated separately. The latter by defaults covers the SPI annealing instants and a monthly variability. This is the best choice for point sources. We suggest not to touch the other parameters of the window, unless a specific analysis is pursued.

6.8 Phase Resolved Analysis

Starting with OSA 6.0, the spi_science_analysis script offers phased-resolved analysis capabilities. The main switch is in the “Histogram options” of the top-level GUI. Clicking on this button opens a window shown in Fig. 27. This feature is not implemented in spimodfit_analysis.

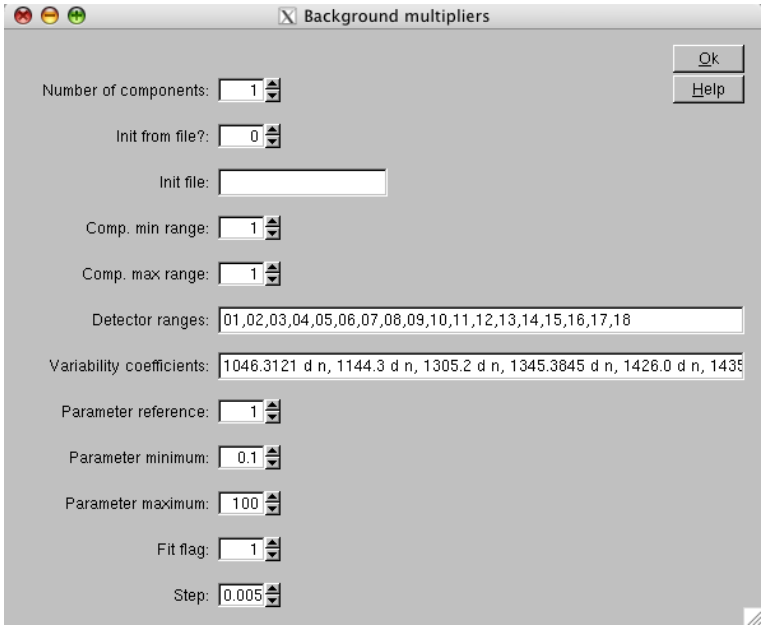


Figure 25: GUI of `spimodfit_analysis` regarding the *spimodfit* background multipliers.

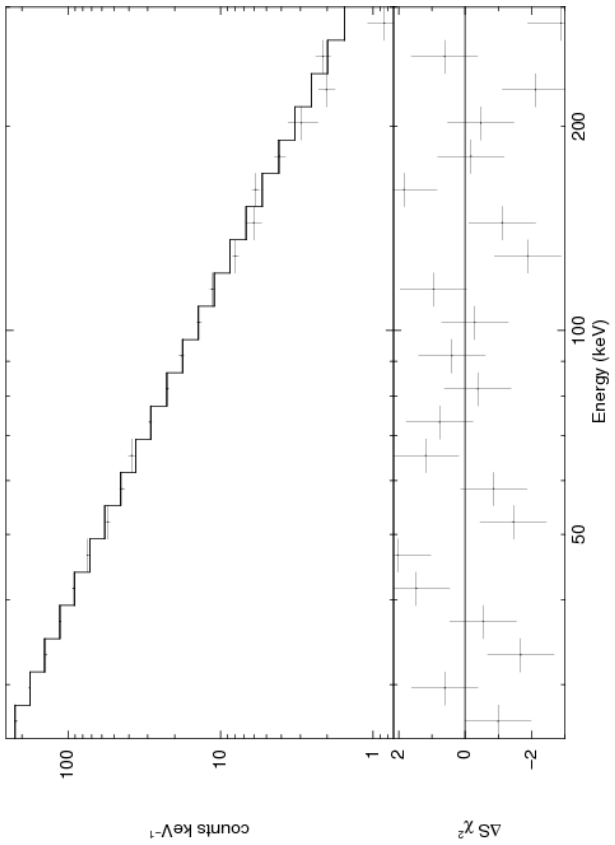


Figure 26: Count rate spectrum of the Crab from the cookbook example and the residuals from the best fit model.

Figure 27: Histogram options GUI

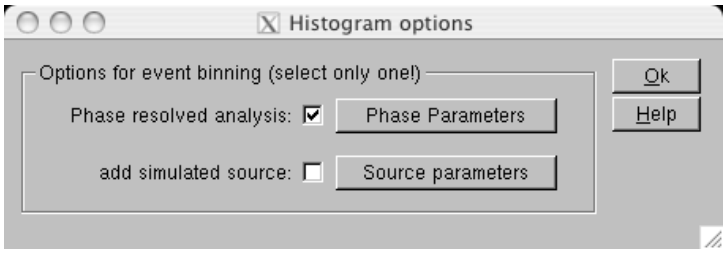
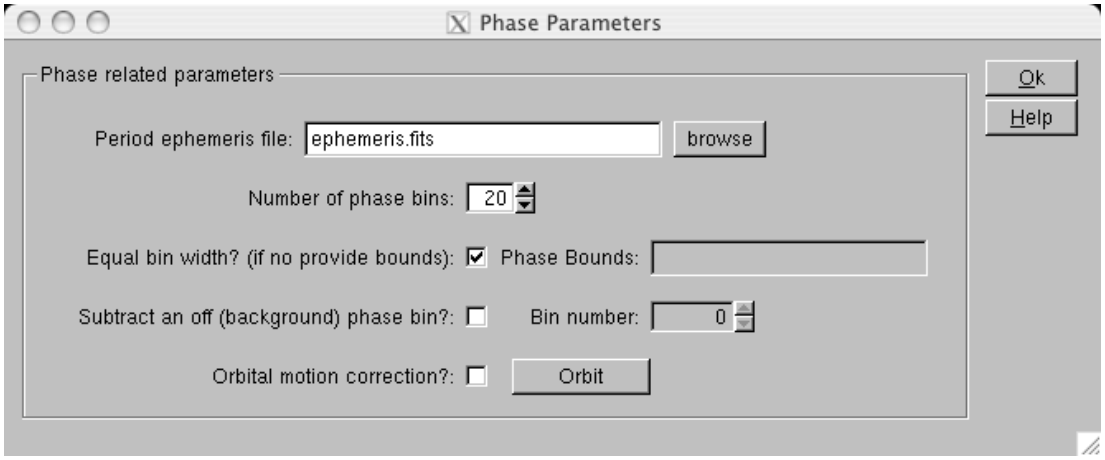


Figure 28: Phase parameter GUI



Open the "Phase Parameters" window (Fig. 28) by selecting "Phase resolved analysis" and clicking on "Phase Parameters". The pulsar ephemeris has to be provided through a FITS file, the format of which is described in the next sub-section (Sect. 6.8.1). You can then specify the number of phase bins. Equally spaced bins will be used by default, but if you un-select "Equal bin width" you can provide specific phase bounds. Be careful when entering this parameter as it may seem confusing. Here is an example for the Crab of a phase bound vector with "Number of phase bins = 7":

Phase Bounds = 0.00 0.03 0.20 0.38 0.50 0.80 0.98 1.00

"Phase Bounds" must include values between 0 and 1, and it must start with 0 and end with 1. However, the first bin (0 to 0.03) and the last bin (0.98 to 1.00) will be merged in the program, resulting in a phase bin -0.02 to 0.03, or equivalently 0.98 to 1.03 one period later. Therefore, the above example is a 6 phase bin analysis despite the "Number of phase bins = 7". The number and boundaries of the phase bins are clearly written in the logfile, that you should consult in case of any doubt (see below).

The following parameter "Subtract an off phase bin" allows to specify an off-pulse phase bin number. The counts from the off-pulse phase bin will be subtracted to the counts of all other bins, after being properly scaled to their actual phase bin sizes. The phase bin are numbered starting with 0. This option can be used with regularly or irregularly spaced bins, although in general an off-pulse phase bin will have to be defined using specific bounds. Do not forget that in the case the first and last bins are merged (see above).

Below is the log file output for the above case. Again the parameter "Number of phase bins" is 7 in this example, 8 "boundaries" are provided, but it is a 6 phase bin case in reality (numbered from 0 to 5). In this case, "Subtract an off phase bin" was also selected with a "Bin number" equal to 4.

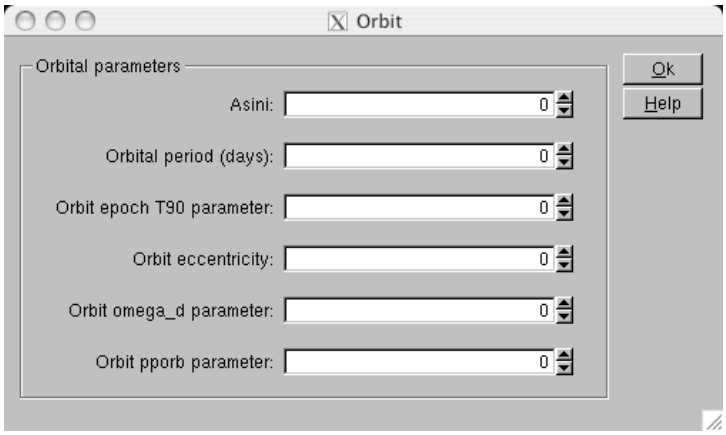
```
Number of phase bin ..............: 6
Phase bin No  0 : from 0.000000 to 0.030000 and from 0.980000 to 1.000000
Phase bin No 1 : from 0.030000 to 0.200000
Phase bin No 2 : from 0.200000 to 0.380000
Phase bin No 3 : from 0.380000 to 0.500000
Phase bin No 4 : from 0.500000 to 0.800000
Phase bin No 5 : from 0.800000 to 0.980000
Counts from phase bin 4 will be subtracted from all spectra
```

Another example: if you select "Equal bin width?", the parameters are less confusing, and with a "Number of phase bin = 5" the log file contains the following outputs.

```
Number of phase bin ..............: 5
Phase bin No 0 : from 0.000000 to 0.200000
Phase bin No 1 : from 0.200000 to 0.400000
Phase bin No 2 : from 0.400000 to 0.600000
Phase bin No 3 : from 0.600000 to 0.800000
Phase bin No 4 : from 0.800000 to 1.000000
```

Finally, an orbit can be specified in the case of a pulsar with a companion (X-ray binary) by entering orbit parameters in the "orbit" window (Fig. 29). Leave zeros if there is no orbit.

Figure 29: Orbit parameter GUI



The phase-resolved analysis is then executed after you have clicked three times on "Ok" buttons. The binning step produces one set of detector spectra for each phase bin, called "evts_det_spec_phase_i.fits" where i is the phase bin number. Note that these files named "spec" or "spectra" are our particular format to store binned events but that these "count spectra" may comprise only one energy bin in case of imaging analysis.

The next background step is phase bin independent (since the background is eventually scaled to the data in the image reconstruction), and consequently it is executed only once. Spiros is run independently for each phase bin in turn and one set of results is produced for each bin. Again spectra and their index and numbered with introducing a "phase_i" to names used for phase independent analyses, where again i is the phase bin number. The same RMF response file (spectral_response.rmf.fits) is valid for all phased-resolved spectra (and automatically attached to them), and therefore, the XSPEC spectral fitting of these spectra is straightforward.

6.8.1 Input ephemeris file

Ephemeris is provided to the program through a FITS file because in the case of the Crab, one set of ephemeris parameters per month is required to follow the Crab phase over periods much longer than one month. The FITS table includes one ephemeris parameter set in each row with a validity time interval (specified through MJDSTART and MJDSTOP). The program can then automatically select a valid ephemeris parameter set (in a given row) for each processed science window. You just have to make sure that the validity intervals encompass the time span of all data included in the analyzed observation group.

We are aware that using a FITS file may be inconvenient for most X-ray pulsars for which only one set of ephemeris parameters is needed. For the time being, this set has to be provided through a FITS table with a unique row with a very wide validity interval covering the whole INTEGRAL mission.

Ephemeris files for the Crab, V0332+53, and 3A0535+262 can be found in

http://isdc.unige.ch/Instrument/spi/download/pulsar_ephemeris/Crab_ephemeris.fits

http://isdc.unige.ch/Instrument/spi/download/pulsar_ephemeris/V0332+53_ephemeris.fits

http://isdc.unige.ch/Instrument/spi/download/pulsar_ephemeris/3A0535+262_ephemeris.fits

The meaning of the ephemeris parameters are standard, information can easily be found trough the WWW, for the Crab in particular, look at

http://www.jb.man.ac.uk/ pulsar/crab.html

For other not so fast pulsars, some of the parameters such as second derivative can be set to zero.

6.9 Gamma Ray Burst imaging

In the following cookbook section we explain how to derive images of short bursts, such as GRB, from SPI data. We assume that readers have some knowledge about SPI data analysis with the ISDC spi_science_analysis script. Please follow the first cookbook example if it is not your case.

1. The first step is to record the start (grb_start) and stop (grb_stop) times of the period you want to use to derive the image, and possibly the time before (sec_to_avoid_before_grb) and after (sec_to_avoid_after_grb) the burst that should not be included in the period used to derive the background. Times can be in UTC or IJD. The best way to look at the timing is to display the burst light curve, for example from IOSM. Of course, you can repeat the analysis in as many time bins as you wish, e.g., to derive a set of images and a SPI light-curve. Figure 30 provides an illustration of the required times on top of a real GRB light-curve.
2. Set up your analysis environment as indicated in the "Setting up the analysis environment" section of the SPI data analysis cookbook.
3. Identify the pointing encompassing the burst, and use og_create to build an observation group (only analysis of single pointing are possible in this frame), as shown in Fig. 31.

   This produces an observation group with a standard name "og_spi.fits" (located in ./obs/grb030501/ in our example).
4. Move to the observation group location (cd obs/grb030501 in our example).
5. Two alternatives can be used to specify the parameters of your analysis. The simplest is to use spi_science_analysis GUI.
   (a) Run "spi_science_analysis" and specify parameters through the GUI, as displayed in Fig. 32.

Figure 30: Illustration of the different GRB start/stop times

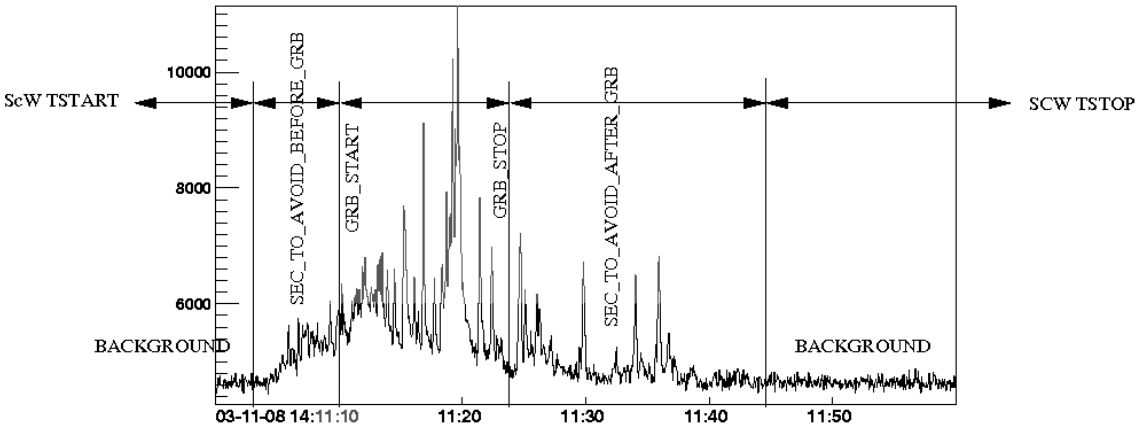


Figure 31: Output of og_create on the terminal

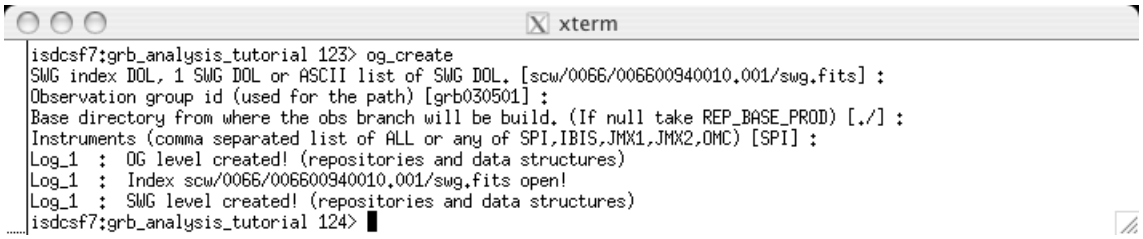
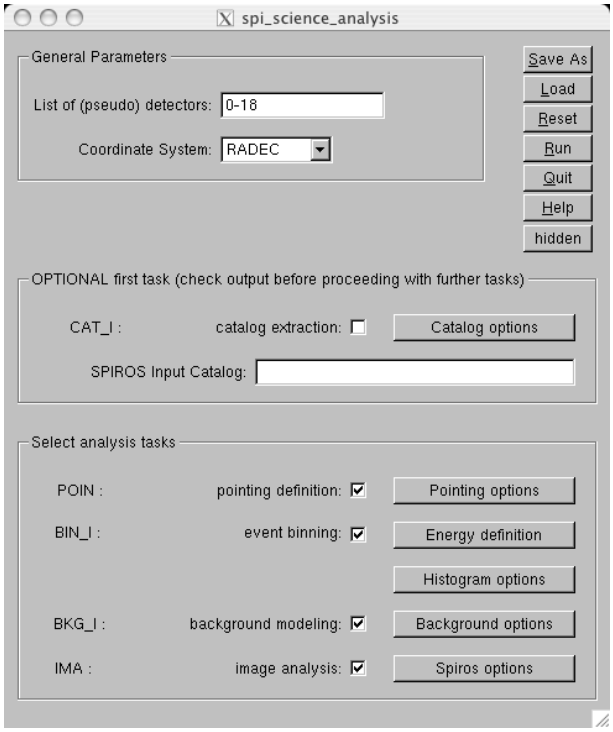

xterm
isdcsf7:grb_analysis_tutorial 123> og_create
SWG index DOL, 1 SWG DOL or ASCII list of SWG DOL. [scw/0066/006600940010.001/swg.fits] :
Observation group id (used for the path) [grb030501] :
Base directory from where the obs branch will be build. (If null take REP_BASE_PROD) [./] :
Instruments (comma separated list of ALL or any of SPI,IBIS,JMX1,JMX2,OMC) [SPI] :
Log_1 : OG level created! (repositories and data structures)
Log_1 : Index scw/0066/006600940010.001/swg.fits open!
Log_1 : SWG level created! (repositories and data structures)
isdcsf7:grb_analysis_tutorial 124>

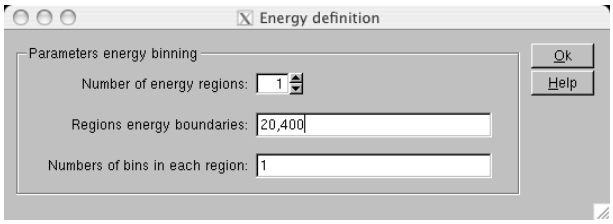
Figure 32: Main GUI window of the spi_science_analysis script



(b) Make sure that the “catalogue extraction” is un-selected, and that no “SPIROS Input Catalogue” (in this imaging example we assume that the GRB position is unknown).

(c) Check the ”List of detectors” (use 0-18 in most cases) and the “Coordinate System” in that window and click on ”Energy_definition” to open the GUI window displays in Fig. 33.

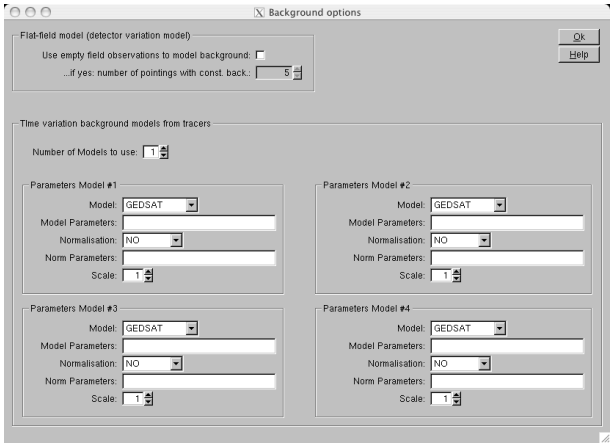
Figure 33: Energy boundaries (or ebounds) GUI



(d) Select your energy bins (in principle a single wide band for a first image), click ”Ok” to close this window.

(e) Click on “Background options” and un-select the flat-field option and select the “Use tracers to model background” option as shown in Fig. 34, and click ”Ok” to close this window.

Figure 34: Background GUI



(f) Back in the parent window click ”spiros” to open the GUI window shown in Fig. 35.

(g) Make sure the “imaging” mode is specified, do not introduce any selection parameters, and specify the ”Background method” and the ”Optimization statistic”. Background method ”1” means that the background derived from the periods before and after the burst, and rescaled to the burst duration will be subtracted from the burst data as a fixed background. With background method ”3” the background will be scaled through the image deconvolution process, solving for a single background scaling coefficient. Do not use other options. For the ”Optimization statistic”, Chi2 is recommended, while LIKEH can be tried in a second run. Click on ”imaging” to check that the default parameters shown in Fig. 36 are correctly entered (and update them if necessary).

(h) Close all sub-windows by clicking twice ”Ok”, then using the ”Save As” button, you can save the parameters you have just entered into a file. Enter the name “grb_analysis.par”[5] and store it in the current directory. PLEASE MAKE SURE NO SPELLING MISTAKE

[5]Note that although this file has a .par extension it is not a standard parameter file. It is a csh script containing all parameter values. The value of all parameters are then later extracted from this file in the main grb analysis script.

Figure 35: spiros GUI main window

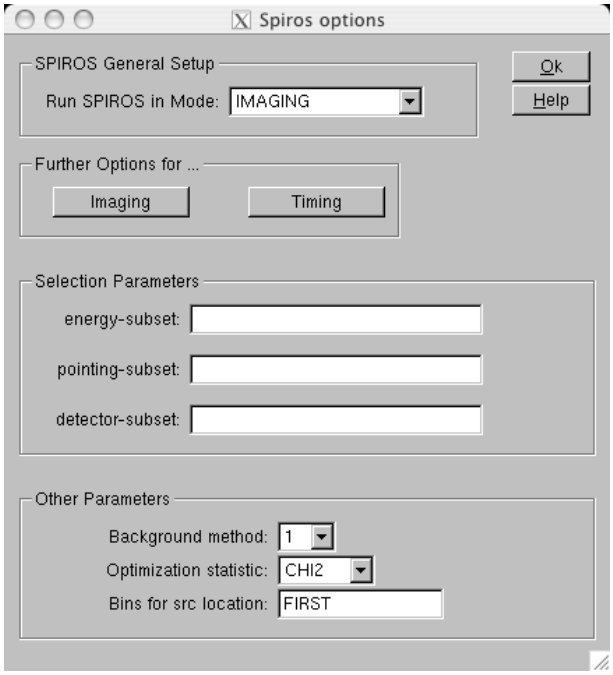
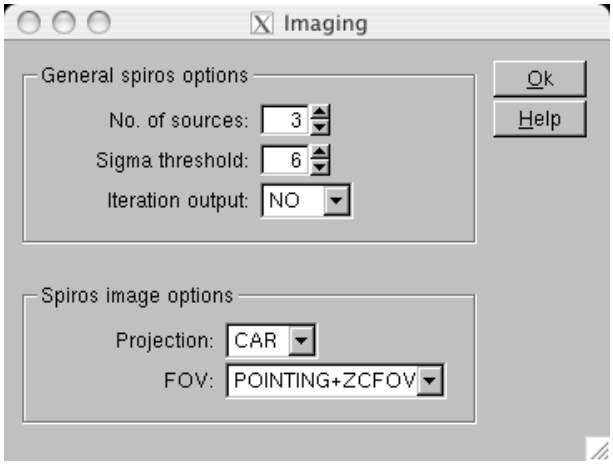


Figure 36: spiros GUI imaging window



IS INTRODUCED. THE GRB SCRIPTS READS THE “grb_analysis.par” FILE LOCATED IN THE CURRENT DIRECTORY TO LOAD THE ANALYSIS PARAMETERS.

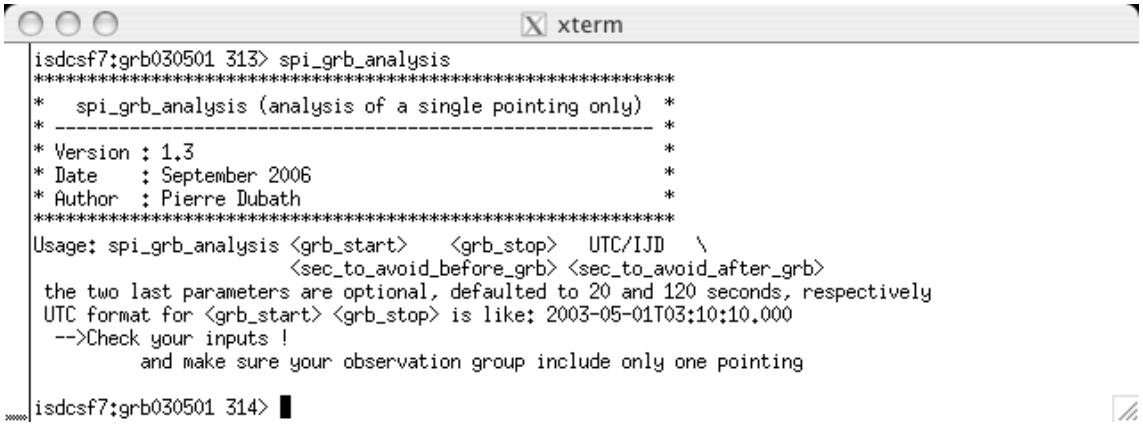
(i) Once the parameters are “saved”, click on “Quit” to quite the process. We do not want to run the analysis yet. This procedure is just a convenient way to enter parameters for the later analyses.

Alternative: If you do not want to use the GUI, you can specify the above parameters by editing the spi_science_analysis.par file located in your $PFILES directory.

6. Now, the “grb_analysis.par” file located in the current directory now contains the main parameters you want to use in your burst analysis. This file will be used by the script.

7. Run “spi_grb_analysis”. Without any command line argument you get the following help text.

Figure 37: Outputs of spi_grb_analysis when invoked without any input parameters



You know the input parameters from the first point above, so you can then launch the spi_grb_analysis script. Make sure the ”grb_start” - ”sec_to_avoid_before_grb” is not before the pointing start, and that ”grb_stop” + ”sec_to_avoid_after_grb” is not after the pointing end, as there is no corresponding checks in the spi_grb_analysis script.

In our example:

spi_grb_analysis 2003-05-01T03:10:10.000 2003-05-01T03:10:30.000 UTC 10 10.

You can re-run spi_grb_analysis in the same directory as many times as you wish, changing the parameters as explained in above section 5. However, when making a new run, the outputs of the previous run will be deleted. If you want to keep your outputs before making a new run save them by copying the entire directory (e.g., cd ..; cp -r grb030501 grb030501_imaging; cd grb030501) before continuing.

8. ”spi_grb_analysis” is calling a number of executables and ”spi_science_analysis” several times. Since ”spi_science_analysis” produces its own log file, the log file situation after running ”spi_grb_analysis” is somewhat complicated. The resulting ”spi_grb_analysis.log” is the high level log file, containing logs from the executables called directly by ”spi_grb_analysis”. In addition, you can find three log files “spi_pointing.log”, “spi_binning.log”, and “spi_spiros.log”. The first contains the outputs of spi_obs_point, the second includes the logs from the binning to background steps, and the third contains the results from spiros.

9. After a successful execution of ”spi_grb_analysis”, it is also possible to run spiros again with the ”spi_science_analysis” script. Launch ”spi_science_analysis”, make sure you select only the spiros step (if you select any other step, the burst background produced by ”spi_grb_analysis” will be deleted and you will have to start it all again), change the required spiros parameters (making a LIKEH analysis rather than a Chi2 one for example), and run the show again.

10. Please consult the cookbook for information about the analysis results (and for more information about the analysis parameters).

6.10 Gamma Ray Burst spectroscopy

In the following cookbook section we explain how to derive spectra of short bursts, such as GRB, from SPI data. We assume that readers have some knowledge about SPI data analysis with the ISDC spi_science_analysis script. Please follow the first cookbook example if it is not your case.

Please follow points 1 to 4 from the previous section 6.9 on GRB imaging. Once this is achieved, you have an observation group in the current directory and you can continue with the following steps.

1. The burst position at which you want to extract a spectrum must now be specified. This is done by providing a "source_cat.fits" file containing the GRB position. You can either use an existing file, or create a new one using the spi_science_analysis to run the catalogue extraction step (the spi_science_analysis main GUI is shown in Fig. 32).

   Select "catalogue extraction" and unselect all other tasks, click on "catalog" to enter the catalogue task parameters. Click "Ok" to close the "catalogue" window, and on "Run" in the main frame to execute spi_science_analysis. The "source_cat.fits" catalogue produced contains irrelevant data. It will probably includes a number of catalogue sources that may be detected on longer observation, and certainly not the GRB position at which we want to extract a spectrum. The catalogue file should include a single row with the GRB identification and position. Edit this file with "fv" and enter the correct identification and position, deleting extra rows if any, and save the result.

   Alternatively, the position from the SPI image analysis can be used (see Sect. 7.19 for how to transform a "source_res.fits" into a "source_cat.fits"), although in most cases a more accurate position should be available from IBIS or other sources.

   If you use an existing "source_cat.fits" file make sure with fv that it contains the correct identification and position of the GRB and that the ISDCLEVL keyword of this file is CAT_I.

2. As in the previous GRB imaging analysis case, two alternatives can be used to specify the parameters of your analysis. The simplest is to use spi_science_analysis GUI.

   (a) Run "spi_science_analysis" and specify parameters through the GUI, as displayed in Fig. 38.

   (b) Check the "List of detectors" (use 0-18 in most cases) and the "Coordinate System" in that window, de-select "catalogue extraction", enter the name of your file (source_cat.fits) in the "SPIROS Input Catalog" entry field and click on "Energy_definition" to open the GUI window displays in Fig. 39.

   (c) Select your energy bins (in this case 6 bins ranging from 20 to 200 keV, logarithmically spaced as the bin number is negative), click "Ok". This will close this window.

   (d) Click on "Background options" and un-select the flat-field defaults option as shown in Fig. 34, and click "Ok" to close this window.

   (e) Back in the parent window click "spiros" to open the GUI window shown in Fig. 40.

   (f) Make sure the "SPECTRA" mode is specified, do not introduced any selection parameters, and specify the "Background method" and the "Optimization statistic". Background method "1" means that the background derived from the periods before and after the burst, and rescaled to the burst duration will be subtracted from the burst data as a fixed background. With background method "3" the background will be scaled through the image deconvolution process, solving for a single background scaling coefficient. Do not use other options. For the "Optimization statistic", Chi2 is recommended, while LIKEH can be tried in a second run.

Figure 38: Main GUI window of the spi_science_analysis script

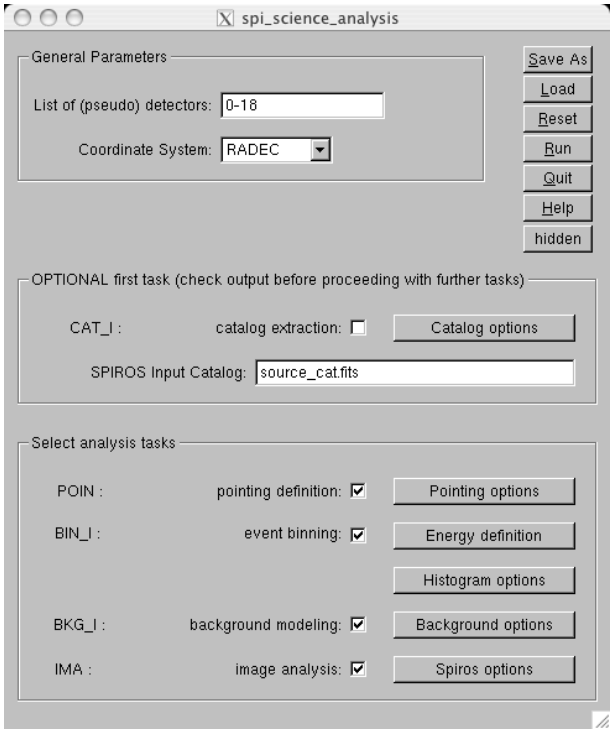


Figure 39: Energy boundaries (or ebounds) GUI

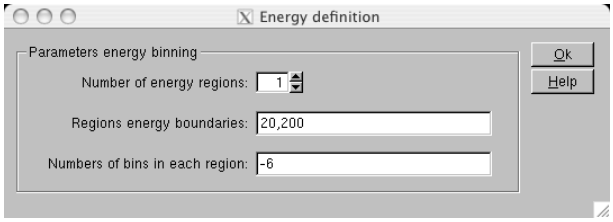
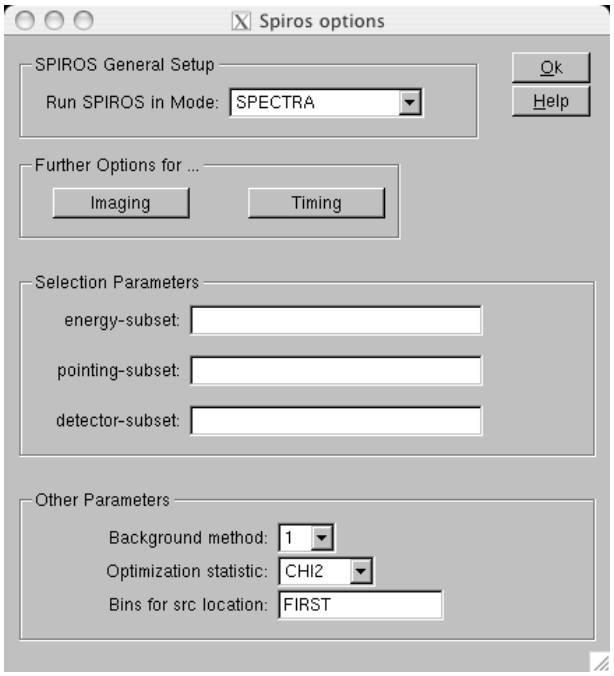


Figure 40: spiros GUI main window



(g) Close all sub-windows by clicking twice "Ok", then using the "Save As" button, you can save the parameters you have just entered into a file. Enter the name "grb_analysis.par" and store it in the current directory. PLEASE MAKE SURE NO SPELLING MISTAKE IS INTRODUCED. THE GRB SCRIPTS READS THE "grb_analysis.par" FILE LOCATED IN THE CURRENT DIRECTORY TO LOAD THE ANALYSIS PARAMETERS.

(h) Once the parameters are "saved", click on "Quit" to quite the process. We do not want to run the analysis yet. This procedure is just a convenient way to enter parameters for the analyses.

Alternative: If you do not want to use the GUI, you can specify the above parameters by editing the spi_science_analysis.par file located in your $PFILES directory.

3. Now, the "grb_analysis.par" file located in the current directory now contains the main parameters you want to use in your burst analysis. This file will be used by the script.

4. Run "spi_grb_analysis". Without any command line argument you get the following help text.

   You know the input parameters from the first point above, so you can then launch the spi_grb_analysis script. Make sure the "grb_start" - "sec_to_avoid_before_grb" is not before the pointing start, and that "grb_stop" + "sec_to_avoid_after_grb" is not after the pointing end, as there is no corresponding checks in the spi_grb_analysis script.

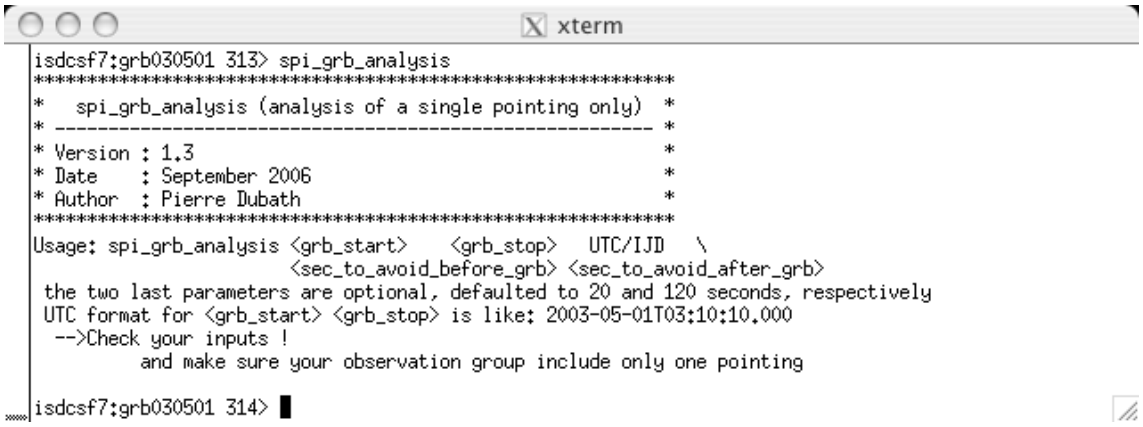
   In our example:

   spi_grb_analysis 2003-05-01T03:10:10.000 2003-05-01T03:10:30.000 UTC 10 10.

   You can re-run in the same directory spi_grb_analysis as many times as you wish, changing the parameters as explain in above section 5. However, when making a new run, the outputs of the previous run will be deleted. If you want to keep your outputs before making a new run save them by copying the entire directory (e.g., cd ..; cp -r grb030501 grb030501_spectra; cd grb030501) before continuing.

5. After a successful run of "spi_grb_analysis" (step 7) you obtain a spectrum of the burst, called spectrum_"your GRB ID".fits, containing a link to the appropriate RMF response.

Figure 41: Outputs of spi_grb_analysis when invoked without any input parameters



You can directly enter XSPEC to display and fit your resulting spectrum (see Sect.7.18 for more information).

6. "spi_grb_analysis" is calling a number of executables and "spi_science_analysis" several times. Since "spi_science_analysis" produces its own log file, the log file situation after running "spi_grb_analysis" is somewhat complicated. The resulting "spi_grb_analysis.log" is the high level log file, containing logs from the executables called directly by "spi_grb_analysis". In addition, you can find three log files "spi_pointing.log", "spi_binning.log", and "spi_spiros.log". The first contains the outputs of spi_obs_point, the second includes the logs from the binning to background steps, and the third contains the results from spiros.

7. After a successful execution of "spi_grb_analysis", it is also possible to run spiros again with the "spi_science_analysis" script. Launch "spi_science_analysis", make sure you select only the spiros step (if you select any other step, the burst background produced by "spi_grb_analysis" will be deleted and you will have to start it all again), change the required spiros parameters (making a LIKEH analysis rather than a Chi2 one for example), and run the show again.

8. Please consult the cookbook for information about the analysis results (and for more information about the analysis parameters).

7 Tips and Tricks

This section gathers a number of important information and advises about the SPI data analysis. We recommend to browse through this section after having exercised the "Cookbook" (Sect. 6), and to come back to one point or another when you are confronted with a particular issue in your own analysis.

7.1 Example of command-line run

The usage of the spi_science_analysis GUI is strongly recommended. It allows to enter all important analysis parameters (see Sect. 7.20 for how to modify the list of important parameters) and then to run the script. However, command-line run are required in some cases, because either (1) spi_science_analysis is called from another, higher-level script, or (2) spi_science_analysis is run through a distant, slow connection, or (3) ROOT has not been installed and thus the GUI is not available.

An easy way to produce a command-line script for spi_science_analysis is to use the GUI to enter the parameters, and then to use the "Save As" button. This produces a few-line sh scripts that you can execute directly or copy into another more sophisticated one.

Below, we also give an example of a command-line run corresponding to the cookbook example of Sect. 6.3. It includes all spi_science_analysis parameters. The example can be "copied" and "pasted" into a terminal or a shell script. Note that this example includes all possible parameters, and that many of them are actually not used in this particular case, but they are still listed such that you can modify them easily.

```
spi_science_analysis obs_group="og_spi.fits" \
    IC_Group="../../idx/ic/ic_master_file.fits[1]" \
    IC_Alias="OSA" \
    coeff_DOL="" \
    IRF_DOL="" \
    RMF_DOL="" \
    catalog="" \
    clobber=yes
    log_File="spi_sa.log" \
    run_cat_extract=no \
    run_pointing=yes \
    run_binning=yes \
    run_background=yes \
    run_simulation=no \
    run_spiros=yes \
    run_phase_analysis=no \
    run_gaincorrection=no \
    run_fullcheck=no \
    detectors="0-18" \
    spiros_source-cat-dol="" \
    coordinates="RADEC" \
    cat_extract_fluxMin="0.001" \
    cat_extract_fluxMax="1000" \
    use_pointing_filter=yes \
    spibounds_nregions=1
    spibounds_regions="20,400" \
    spibounds_nbins="1" \
    spi_phase_hist_ephemDOL="ephemeris.fits" \
    spi_phase_hist_phaseBinNum=20 \
```

```
spi_phase_hist_phaseSameWidthBin=yes
spi_phase_hist_phaseBounds="" \
spi_phase_hist_phaseSubtractOff=no
spi_phase_hist_phaseOffNum=0 \
spi_phase_hist_orbit=no \
spi_phase_hist_asini=0 \
spi_phase_hist_porb=0 \
spi_phase_hist_T90epoch=0 \
spi_phase_hist_ecc=0 \
spi_phase_hist_omega_d=0 \
spi_phase_hist_pporb=0 \
spi_add_sim_SrcLong=80 \
spi_add_sim_SrcLat=19 \
spi_add_sim_FluxScale=0.01 \
use_background_flatfields=yes \
use_background_templates=no \
use_background_models=no \
spi_flatfield_ptsNbConstBack=5 \
spi_flatfield_single=no \
spi_templates_type="GEDSAT" \
spi_templates_scaling=1 \
spi_obs_back_nmodel=1 \
spi_obs_back_model01="GEDSAT" \
spi_obs_back_mpar01="" \
spi_obs_back_norm01="NO" \
spi_obs_back_npar01="" \
spi_obs_back_scale01=1 \
spi_obs_back_model02="GEDSAT" \
spi_obs_back_mpar02="" \
spi_obs_back_norm02="NO" \
spi_obs_back_npar02="" \
spi_obs_back_scale02=1 \
spi_obs_back_model03="GEDSAT" \
spi_obs_back_mpar03="" \
spi_obs_back_norm03="NO" \
spi_obs_back_npar03="" \
spi_obs_back_scale03=1 \
spi_obs_back_model04="GEDSAT" \
spi_obs_back_mpar04="" \
spi_obs_back_norm04="NO" \
spi_obs_back_npar04="" \
spi_obs_back_scale04=1 \
spiros_mode="IMAGING" \
spiros_energy-subset="" \
spiros_pointing-subset="" \
spiros_detector-subset="" \
spiros_background-method=3
spiros_srclocbins="FIRST" \
spiros_image-proj="CAR" \
spiros_image-fov="POINTING+ZCFOV" \
spiros_nofsources=3 \
spiros_sigmathres=6 \
spiros_iteration-output="NO" \
spiros_optistat="CHI2" \
spiros_source-timing-mode="QUICKLOOK" \
spiros_source-timing-scale=0.1
```

7.2 Repeating and saving analyses

As data access is carried out through the observation group (OG), all the "spi_science_analysis" script needs to know about the input data is the default name "og_spi.fits"[6]. All output data files are created with default standard names and they are attached to the OG, so that subsequent tasks can access them through the OG.

It is always possible to repeat an analysis starting from any step. Simply select the tasks you want to execute by clicking on the boxes in front of the "spi_science_analysis" buttons. Be careful however, to define a coherent analysis. In other words, do not leave gaps in the succession of the tasks. In such cases the "spi_science_analysis" would stop immediately with a "Incoherence in the succession of tasks" error.

A concept of analysis level hierarchy has been defined, as indicated in Table 3. When the pipeline processing proceeds from one step to another, all the data are stamped with the current level. A fresh initial OG starts with the "DEAD" level, and then goes trough the "POIN", "BIN_I", "BKG_I", and "IMA" (SPE or LC) levels successively in a complete analysis. When "spi_science_analysis" is launched, any data with a stamped level "higher" than the new starting level is deleted before the actual processing starts again. As a consequence, the outputs from a previous analysis are always overwritten.

The easiest way to "save" the results of a previous analysis is to copy the complete directory structure before starting again.

```
cd ..
cp -r analysis_current_dir analysis_saving_dir
cd analysis_current_dir
```

As all analysis data are part of the OG, you cannot use "rm" to delete some of them. This would break the OG and prevent any further (re)processing. If you use "rm" by mistake, the safest is to start again at the very beginning with og_create. If you want to delete some data from the OG without using the "spi_science_analysis" script, you can use the *spi_clean* cleaning tool described in Sect. 8.10.

7.3 Pseudo-detectors and detector list

Incoming photons which hit more than one SPI detector are called multiple events. A set of "pseudo-detectors" are defined to incorporate double and triple events into the analysis. Simulations showed that events hitting two detectors originate from a relatively small area (smaller than that of a detector) located near the border of the two detectors. The same is true for triple events, they occur very near the common corner of the three involved detectors. Hence the idea of considering these small areas located between detectors as "pseudo detectors". The response of these "pseudo detectors" is computed from GEANT simulations, and they are included in the analysis as if they were real detectors. Events hitting more than 3 detectors are increasingly dominated by background events. They are not normally used in scientific analysis.

In general for analysis up to a few hundreds of keV, the inclusion of the double and triple events does not significantly improve the results. In other words, for point sources, most analysis can be restricted to "real detectors", i.e, a detector list 0-18. It is probably always a good idea to start an analysis first using only real detectors anyway. For high-energy analysis, double (pseudo-detectors 19 – 60) and triple (pseudo-detectors 61 – 84) can be added later into the analysis, using an extended detector list. Note however that the current RMF response available for XSPEC spectral response is valid for analysis with real detectors (0-18) only.

[6] or an alternative OG name if you do not follow the standards

7.4 Catalogue extraction

Although it is possible to carry out an image reconstruction without any prior information, better results will be obtained if the positions of sources known to lie in the field-of-view are specified beforehand. In any case, source positions have to be provided for spectral extraction.

Source positions are provided through a source_cat.fits file. For a specific analysis, you can either (1) use an existing source_cat.fits, or (2) perform an extraction (cat_extract = CAT_I = first step of the analysis) from the ISDC reference catalogue. Since the outputs of cat_extract are sometime hard to predict, it is strongly advisable to run only cat_extract first and to check the result with the ftool "fv" before going further in the analysis. If there are sources that should not be taken into the analysis, either set the SEL_FLAG column (one before the last one) to zero, or removed the un-wanted rows. Alternatively, make sure SEL_FLAG is set to one for all sources that should be used in the analysis.

In general, it is difficult to know in advance which of the known sources will actually be detected by *INTEGRAL* . It is therefore a good practice to make an analysis without specifying any source positions, to look at the results, possibly comparing them with the outputs of cat_extract, before evaluating which source positions should be fixed in the analysis.

7.5 Event energy correction

Since the start of the second revision of the ISDC processing, sets of gain coefficient files are part of the analysis input data. For each revolution, a dedicated ISDC pipeline (1) builds spectra from all available events, (2) derives the positions of a number of background lines, and (3) compute gain coefficients by fitting a polynomial to the line position – energy relation (see Sect. 8.2 for more details). The "spi_science_analysis" pipeline uses these files by default for on-the-fly energy correction, in the event binning step (program spi_obs_hist).

This "standard" energy correction is excellent for most cases, even for line analysis (only line analysis in revolutions with large temperature changes may require better correction). You can however, if you want, apply your own set of gain coefficients by providing the name of an alternative index of gain coefficient files in the "Gain Coefficient file name" entry field of the spi_science_analysis GUI (click on the upper right "hidden" button to find this parameter).

7.6 Event binning, GTI and dead times

In addition to the actual event binning, this step includes the energy bin definition (through the spibounds software), the derivation of the GTI and of the dead time (GTI and DEAD tasks). So far no real GTI filter has been implemented at this stage and the complete time intervals of all pointings are considered as "good" (remember that bad pointing can be eliminated during the pointing step using a list of known bad pointings). The actual dead times are provided by the SPI electronic.

Dead time ratio and live time are defined by LIVETIME = DEADC * ONTIME, where the ontime is the net observing time, excluding any gaps in the data taking period, such as gaps due to slews or data loses, and the LIVETIME is the time corrected from the dead time

These definitions are borrowed from

http://legacy.gsfc.nasa.gov/docs/heasarc/ofwg/docs/ofwg_recomm/r11.html

7.7 Background modeling

In the most general case, the SPI image reconstruction is an intractable problem as it would require solving for one background value for each combination of detector, pointing, and energy bin in addition to solving for the source parameters. This problem has more unknowns than measurements. Clearly some assumptions on the background behavior have to be introduced.

For all point source analyses, we recommend to use the default flat field option. In this case, predefined flat-field spectra are used. They are rebinned to match the bin size of your analysis, and the program will select and use the appropriate flat-field spectrum for your analysis automatically. With this option, the detector to detector ratios are kept fixed (from that of the empty field), but one background coefficient is derived through the imaging reconstruction process. In fact, one coefficient is derived for each time interval, and the length of the intervals is provided as a number of consecutive pointing through the next parameter. We use a value of 5 in the cook book example. The background is then assumed to be constant within these intervals of 5 pointings. Note that in general, much longer time interval should be used in the range 20 to 100 pointings.

A good background modeling alternative is provided by either the GEDSAT template or the GEDSAT model, which are based on the count-rate of the events saturating the Germanium detector electronic. In this case, un-select the flat-field option and select the corresponding background option. For the GEDSAT model set the "Number of Models to use" to 1, and select the GEDSAT model. In the image reconstruction process, we will assume that the real detector background varies in the same way as this model, which will be scaled to the actual data to determine the real background. GEDSAT is a good model for most point-source analyses.

It is recommended to read further advises from the user manuals of the spi_obs_back and SPIROS programs before thinking about using more sophisticated background models. See also the documents provided by the SPI team [10] for difficult cases.

7.8 Image reconstruction

Image reconstruction is carried out using SPIROS in IMAGING mode. It is recommended to first use background method 3 in conjunction with the flat-field background model. The CHI2 optimization statistic should first be used, while the LIKEH can be tried in a second run. In principle, both optimization statistics should produce compatible results. However, with the CHI2 statistic all the bins with zero counts are ignored, and this can lead to biases (hard tail at high energy, or problem with pseudo-detectors at low energy where the number of double/triple can be null). On the other hand, CHI2 statistic seems to be more robust and faster in most cases.

SPIROS searches for the brightest source first, it then removes its contribution to the original binned event data set, and continues iteratively with increasingly fainter sources. The maximum number of sources SPIROS should search for must be specified as well as the minimum acceptable sigma for source detection. (The number of sources is meant to be in addition to those that may be specified in a source_cat file). For the source number, try first with the number of sources that should be detected in your *INTEGRAL* data set given the field of view, plus 1 or 2 for possible unexpected sources. For the sigma threshold, use 5 or 6 to be safe. Below that level you may get spurious detections as SPIROS is a bit optimistic at the lower S/N end.

Since SPIROS reconstruct a sky image, the field-of-view (FOV) has to be specified. A convenient way to do that is to use the FOV parameter. With POINTING+ZCFOV, SPIROS will always image a square angular region defined by the extent of the centers of all pointings augmented by the SPI zero-coded FOV (see below Sect. 7.9 for a possible strong bias).

Sky images are produced at each iteration. The final images produced after the last iteration are "spiros_image_intensity_result.fits" and "spiros_image_sigma_result.fits". The source positions and fluxes are output in source_res.fits.

7.9 Biases from bright sources "just outside the image FOV"

Because the output sky image is reconstructed over a user defined field-of-view (FOV), it may happen that a bright source located in the instrument FOV is not included in the image FOV. For example, if the image FOV is specified as POINTING+FCFOV (i.e., pointing center + SPI fully-coded FOV), there might be a bright source in a corner, outside of the fully-coded FOV, but inside the zero-coded FOV. A large fraction of the detector events can then come from this source, and SPIROS searches for a source location to best reproduce these counts. SPIROS however cannot find the real source as the search is restricted to the specified sky image. It will be forced to find one (or several) spurious positions, at some coding or random noise peak, inside the image FOV, and it will attribute the counts to it. This will lead to completely biased results.

In order to avoid such biases, first make sure all detectable sources in and close to the FOV of SPI fall inside the specified image FOV. In general, it is enough to use the "POINTING+ZCFOV" (pointing+zero-coded FOV) option in the "spi_science_analysis" FOV entry field (click "spiros" first in the main GUI, and "imaging" in the spiros GUI). If really necessary you can defined your own custom image FOV (see the SPIROS user manual in such a case).

Second, check that SPIROS "maximum number of sources" is not too restrictive. You will also obtain biased results if there are 3 bright sources in your sky image and that SPIROS is limited to finding the two brightest sources. Remember that SPIROS "maximum number of sources" parameter is meant in addition to those specified in the catalogue. Once you have explored your data set, it is a good practice to put all well detected sources in the catalogue (possibly with precise positions from the reference catalogue), and to ask SPIROS to search for 1-2 additional sources. You may then check whether these additional sources are real or not, for example by cutting your data set into two independent ones, and checking if these additional sources are found at the same position in the two independent analyses.

7.10 Spectral extraction

In SPECTRAL mode, SPIROS simply needs to know the source positions (provided through the source_cat file) at which to extract spectra. The only important parameters in this step are the background method and the optimization statistics, and the remarks made before for image reconstruction do apply here as well. An RMF response is generated and attached to the derived spectra so that they can directly be used in XSPEC for spectral fitting. Extracted spectra are called "spectrum_*sourceID*.fits".

7.11 Variable sources and timing analysis

A strong limitation of the SPIROS imaging and spectral mode lies in the underlying assumption that the source flux is constant over the whole observation period. If this assumption is wrong and a relatively bright source does vary significantly, the results obtained in these modes will be biased.

One possible indication that a source flux change with time is when the analysis residuals remain large, even after all cares have been taken in the analysis (see next section). In these cases, we recommend to switch to the SPIROS timing modes (imaging-timing to obtain images, timing-spectra to obtain spectra, and timing to get light curves) and to let some of the suspected sources vary during the observation. To tell spiros to let a certain source vary, open your source catalogue with fv and edit the "SEL_FLAG" column and set it to 2 (normally, this column contains 1 one for the sources to be used and 0 for the sources to be ignored by spiros). It is probably best to first let vary only one source (the brightest of the suspects) and to check the residuals again. If the residuals are still too large, try with a second source and so on. But do not forget that, as usual, the best solution will be the simplest model capable of reproducing the observed counts with statistically reasonable residuals. Note that it is a particular bad idea to simply set many (or

all) sources to variable! In such a case, spiros will be unable to derive a reliable solution and the results may be arbitrary.

A detailed discussion with exmaples is provided in two separate documents: "SPI Compact Source Analysis" and "SPI Complex Case Analysis", both available from the Documentation section of the ISDC homepage [10].

7.12 Converging on a good solution - checking residuals

At the end of any SPI data analysis, the SPIROS solution residuals provided in the log file must be checked. DO NOT TRUST any solution with residuals larger than 3 standard deviations. The only possible exception is when considering one of the few cases with relatively high signal-to-noise and signal-to-background ratio, such as the Crab or Vela. In these cases, large residuals may come from non-poissonian errors which are not taken into account by SPIROS.

The first analysis of any dataset will in most cases have too large residuals. The first thing to check is the table provided in the SPIROS log-file containing the solution residual for each pointing. Although bad pointing are filtered in the first (pointing) steps, there might still be a few pointings with large residuals. The next step is to activate the AUTO filter of spiros by entering "AUTO" in the "pointing-subset" entry field. Re-run the pipeline and check the before mentioned table in the log file. It is important that no high residuals remain.

Inappropriate background modeling will also lead to large residuals. Try to improve the background modeling changing time scales and/or methods.

Check the list of detected sources. Large residual can be induced by the presence of bright source located just outside the field-of-view. Another case is when SPIROS is limited to a number of sources (through the No. of sources parameter) and that there is an additional source in the FOV. In this case, SPIROS cannot take this source into account and its counts are attributed to the other sources and/or to the background.

When all the above checks are performed, and if there are still large residuals, it may be time to let varying some sources. SPIROS in imaging and spectra modes assumes that the source fluxes are constant throughout the whole observation period. A good solution can therefore not be obtained if some of the sources are variable (see Sect. 7.11).

Only if bad pointings remain despite all efforts you can remove them from the analysis. Note all bad pointings and enter the GOOD pointings in the pointings-subset entry field (see also Sect. 6.3.1).

7.13 High time resolution with SPI

The highest time resolution possible for light curves is one science window, i.e., about 30 minutes. A higher time resolution is not possible for spiros.

However, it is possible to analyze the raw SPI event data, which have a very high time resolution but no further tools are provided, however, plenty of tools (e.g., the FTOOLS) exist to work on event data.

7.14 SPI failed detectors

Three of the 19 SPI Germanium detector have ceased to function properly and they are now disabled. The failed detectors also affect the response of their neighbors as the double events involving before a neighbor and the "to die" detector are now seen as single events in the neighbor detector (all multiple event are affected similarly). Currently, we have three independent instrumental responses, (1) one for all 19 detectors for the period preceding the first failure, (2) one for 18

detectors for the period between the first two failures, and (3) one for 17 detectors for the period after the second failure and before the third failure. The introductin of a fourth (4) respoonse valid after the failure of the third detector will be released in the next OSA release, after the appropriate testing is performed.

Table 4: SPI detector failures

Det ID	UT	IJD	Pointing	Revolution
#2	2003-12-06T09:58:30	1435.41635	014000040010	140
#17	2004-07-17T11:00:00*	1659.46000*	021400600010-021500000012	214-215
#5	2009-02-19T12:00:00*	3337.50000*	077500610010-077600000012	775-776

* these times are approximation as the failure occurred at some unknown time during the perigee passage.

Our software cannot currently handle a time dependent response, and the easiest way is to analyze the three possible cases independently and to combine the results later on. It is possible however, to analyze different mixtures of data together using one of the three responses as they are not so different. Great care should be taken in this case anyway. For example, because of the effect mentioned in last paragraph, the single event background in the neighbors of a dead detector will increase sharply at the time of the failure, while the other detector backgrounds will remain approximately constant. As a consequence, it will no longer be possible to assume that the background follow the same time evolution in all detectors (as you do when using the saturated Germanium event count-rate as a time model for example).

It is no longer necessary to specifically exclude dead detectors from the analysis. The software does that automatically.

7.15 Annealing phases

About every 6 months, the SPI instrument goes through a so-called "annealing" phase. During these phases, the detectors are heated up to about 100 C for a few days so that their crystal structure can re-arrange and correct itself for most of the accumulated radiation damages. During these phases, listed in Table 5, SPI does not produce any scientific data. At the end of an annealing phase, the detectors are cooled down again and they are switched on before the temperature reached the nominal value. During this period, the standard energy correction is not so good and special care has to be taken for accurate spectral analysis. These periods with unstable temperature are included in the ranges of revolution provided in Table 5. Data from revolutions outside of these ranges should be fine in this respect.

7.16 Instrument modes and SPI data

From the SPI operation point of view, three main different phases can be identified, (1) the first 2 months (pvphase), (2) from the pvphase to the first annealing (approximately revolution 20 to 38), (3) from the first annealing up to the latest revolutions. During the first period, various modes and setups have been tried and the data are a bit more difficult to analyze. During the second one, because of the telemetry budget limitation, all the information from the individual single events could not be downloaded. They were binned on-board and transmitted to ground in the form of spectra instead. The software can handle these spectra in a rather transparent way, although it may be useful to keep this fact in mind in some cases. During the third period starting after the first annealing the full information about all events has been downloaded to the ground and SPI has been operated in the standard scientific photon mode. It may also be useful to know that at the beginning of this period, between the first and the second annealing, data were lost at different rates also due to telemetry limitation. These losses are taken into account in the analysis, and you

Table 5: SPI annealing periods

	UT	IJD	Revolutions included
	start – stop	start – stop	first – last
1	2003-02-06 $\sim 09h$ – 2003-02-14 $\sim 20h$	1132 – 1146	39 – 42
2	2003-07-15 $\sim 12h$ – 2003-07-29 $\sim 18h$	1291 – 1306	92 – 96
3	2003-11-11 $\sim 12h$ – 2003-11-26 $\sim 20h$	1411 – 1426	132 – 136
4	2004-06-17 $\sim 09h$ – 2004-07-02 $\sim 18h$	1629 – 1645	205 – 209
5	2005-01-19 $\sim 09h$ – 2005-02-05 $\sim 06h$	1132 – 1146	277 – 282
6	2005-06-14 $\sim 00h$ – 2005-06-30 $\sim 00h$	1991 – 2008	326 – 330
7	2006-01-09 $\sim 00h$ – 2006-01-26 $\sim 00h$	2200 – 2217	395 – 400
8	2006-06-08 $\sim 00h$ – 2006-06-27 $\sim 00h$	2350 – 2369	446 – 452
9	2006-12-04 $\sim 10h$ – 2006-12-22 $\sim 15h$	2529 – 2547	506 – 511
10	2007-05-29 $\sim 12h$ – 2007-06-17 $\sim 17h$	2705 – 2724	565 – 571
11	2008-01-12 $\sim 04h$ – 2008-01-31 $\sim 07h$	2933 – 2952	641 – 647
12	2008-08-17 $\sim 04h$ – 2008-09-05 $\sim 19h$	3151 – 3171	714 – 720
13	2009-04-20 $\sim 08h$ – 2009-05-11 $\sim 23h$	3397 – 3418	796 – 802
14	2009-10-19 $\sim 18h$ – 2009-11-09 $\sim 9h$	3579 – 3600	857 – 863
15	2010-03-30 $\sim 9h$: – 2010-04-17 $\sim 1h$	3741 – 3759	911 – 916

may notice unusual ratio between the ontimes, which exclude the data loss gaps, and the difference between the GTI stops and starts.

7.17 Crowded fields

When the number of sources in the field-of-view are larger than a few (3-4), the analysis of SPI data becomes increasingly difficult, especially when sources are separated with angles comparable to the SPI resolution of about 2 degrees. SPIROS can get confused and attribute the counts due to one source to another. A good test is to add a mock source (or a few) in the middle of the field and to compare the flux extracted by SPIROS with the input one.

It is possible to derive the contribution of a mock source with the program spi_add_sim. The detector counts expected from a source at given position and intensity are computed using the instrumental response, and added into the binned event data set. This step should be disabled for most standard analysis. The program only works for single energy bin analysis. The user should check the documents provided by the SPI team for further reference [10].

7.18 Fitting SPI spectra using XSPEC

- Extract a spectrum with the spi_science_analysis script following the instructions provided in Sect. 6.5. The spi_science_analysis script sets all relevant parameters in the resulting spectra, creates an appropriate response file, and links it to the spectrum.
- Below is a simple example of an XSPEC spectral fitting session with a Crab spectrum (spectrum_Crab.fits).

```
xspec
cpd /xs
data spectrum_Crab.fits
setpl energy
model power
```

```
2.0
8.0
fit 1000
flux 20,200
plot ldata delchi
```

7.19 Using a SPIROS output catalogue in input (source_res.fits to source_cat.fits

One of the SPIROS outputs is a catalogue (source_res.fits) containing source positions and fluxes. After a SPIROS run you may want to fix the positions of some of the detected sources, and run SPIROS again (see Sect. 6.4), and for that you need to turn an output catalogue(source_res.fits) into an input one (source_cat.fits). Their structures are very similar, therefore, only two keywords and possibly the SEL_FLAG need to be changed as follow

- open the SPIROS output catalogue with fv
- click on 'Table' of SPI.-SRCL-RES
- check that the column SEL_FLAG (third but last column) is 1 for all sources you want to fix in your analysis (source with SEL_FLAG = 0 will be ignored)
- click on 'Header' of SPI.-SRCL-RES
- change EXTNAME to 'SPI.-SRCL-CAT'
- change ISDCLEVL to 'CAT_I'
- save the changes (Alt+S)

7.20 Adding a new parameter in a spi_science_analysis analysis

The spi_science_analysis calls a number of individual programs and feed them with the relevant parameters with a certain logic. Let us take a concrete example, the program spiros, for the sake of clarity. The spiros parameters are listed in spiros.par (found at the $PFILES location. We have selected the most important spiros parameters when building the spi_science_analysis script, and these parameters are named spiros_"parameter's name" in spi_science_analysis.par.

For example, "nofsources" from spiros.par becomes "spiros_nofsources" in spi_science_analysis.par. However, the spiros parameter "kofsources" from spiros.par is not in spi_science_analysis.par, because we decided it was not a parameter user should "normally" change. Remember that users can only directly access the parameters of spi_science_analysis.par, either from the GUI or from the command line.

In the spi_science_analysis code, all parameters are first specified (i.e., hard coded), and both "nofsources" and "kofsources" get default values at this stage. After that the script copies from the spi_science_analysis.par all "spiros_..." parameters, and supersede the default values of these parameters. In this way, the value of "spiros_nofsources" replaces the default value of "nofsources". However, the default value of "kofsources" (Kind OF SOURCES = POINT) cannot be changed as "spiros_kofsources" is missing in spi_science_analysis.par. A nice feature, however, is that it can easily be added.

Take your favorite editor, copy the "kofsources" line from spiros.par into the spi_science_analysis.par and add "spiros_" before "kofsources". Save the spi_science_analysis.par and launch the GUI (make

sure the spi_science_analysis.par edited is at the right $PFILES location). The parameter now appears at the bottom of the main GUI (or at the bottom of the "Hidden" GUI if it is a hidden parameter). The GUI does not look very nice, but you can now modify the value of "spiros_kofsources", and it will be transmitted to spiros as "kofsources". You can also add "spiros_kofsources" on a command-line run.

After your run, you can check the actual value of "kofsources" in the spiros logfile.

7.21 GTIs in SPI data analysis

Current SPI data analyses carried out with the spi_science_analysis script are not making use of any GTIs. In order to eliminate "bad data", only entire pointing are excluded from the analysis (see Sect. 6.3.1. In addition, the GTIs require to analysis short bursts such as GRBs can be automatically by the GRB analysis script. In some cases, you may need to create a GTI, anyway, and this can be done following the below procedure.

- Create an empty GTI structure

  dal_create obj_name="usergti.fits" template="GNRL-USER-GTI.tpl"

- fill some keywords (if you prefer GUIs, use fv)

```
fparkey value=0.0 fitsfile=usergti.fits\[1] keyword=TSTART
fparkey value=10000.0 fitsfile=usergti.fits\[1] keyword=TSTOP
fparkey value=GOOD fitsfile=usergti.fits\[1] keyword=GTI\_TYPE
```

  TSTART and TSTOP define the start and end validity of your selection, not the selection itself. The GTI_TYPE can be 'GOOD' (use the time specified in the user GTI file) or 'BAD' (exclude the time specified in the user GTI file)

- open the fits file with fv

  fv usergti.fits

- click on 'Table'
- click on 'Edit'
- select 'Insert' and then 'row'
- insert as many rows as you want to have good time intervals
- fill in the row either the UTC or the IJD or the OBT columns corresponding to your good/bad time you want to define
- 'Save' the fits file.

8 Data analysis steps

As explained in the Overview Section, the main script calls a number of executables which are launched in a given order. In the present chapter we describe some executables in more details in order to explain how the main script works and what parameters you have to enter for a proper analysis.

Describing the executables we also discuss the parameters used by *spi_science_analysis*. To learn more about a particular executable, type its name with the `--h` option. Non-hidden parameters of the main script are marked with bold font. Usually names of the main script parameters are derived from the corresponding name of the parameters of the executables by adding the name of the executable in the beginning (e.g. parameter `pointing_dol` of the executable *spi_obs_point* has a name `spi_obs_point_pointing_dol` in the main script). In the rare case when this rule is broken we give both names in the description of the executable. A detailed description of the products of each step can be found in the Appendix section, where also the description of the raw data is given.

8.1 Catalog source extraction – CAT_EXTRACT

The executable *cat_extract* performs the source selection from a reference catalog. The output catalog consists of the sources that are in the FOV of the pointings included in the Observation Group. You can also put additional criteria on the sources that would be selected by setting the desirable position (with the parameters `radiusMin` and `radiusMax`, which specifies a surface obtained by merging the areas lying between `radiusMin` and `radiusMax` from the central point of each pointing), and the desirable flux (with the parameters `fluxMin` and `fluxMax`).

Table 6: *cat_extract* parameters included in the main script

Name (main script)	Name (executable)	Type	Description
cat_extract_radiusMin	radiusMin	string	Lower limit for the position selection default: "0"
cat_extract_radiusMax	radiusMax	string	Upper limit for the position selection default: "20"
cat_extract_fluxDef	fluxDef	string	Column used for flux selection possible values: "1" – 15-300 keV "2" – 0.3-8 MeV default: "1"
cat_extract_fluxMin	fluxMin	string	Low limit for flux selection [ph/cm^2/sec] e.g. Crab catalogue value is 0.4458 (15-300 keV) / 0.0167 (0.3-8 MeV) default: "0.001"
cat_extract_fluxMax	fluxMax	string	High limit for flux selection [ph/cm^2/sec] default: "1000"

8.2 Energy correction – SPI_GAIN_CORR

When running the spi_science_analysis script, event energy calculation is carried out on the fly within the spi_obs_hist program. On the other hand, event energies can be computed off-line with the program spi_gain_corr. In the two cases, the corrections are computed in the same way, deriving the energies in physical units (keV) from the instrumental channel (PHA) using a set of

gain coefficients.

The correction can be evaluated with the help of up to five gain coefficients c_i. However, currently four coefficients are used for the low energy range through the following polynomial

$$E = c_0 \times \frac{1}{\mathrm{PHA}} + c_1 + c_2 \times \mathrm{PHA} + c_3 \times \mathrm{PHA}^2,$$

while for the high energy range only the following linear relation is used

$$E = c_0 + c_1 \times \mathrm{PHA}.$$

By default the parameter **randomize** is set to `yes` and the executable performs a randomization of event energies over the instrumental channel width using a uniform random number generator. If **randomise**=`no` then the exact energy of the channel center is attributed to all events of a given PHA channel.

More details can be found in the spi_gain_corr User Manual, which can be downloaded from the following page

http://isdc.unige.ch/Support/spi/doc/doc.html

8.3 Pointing definition – SPI_OBS_POINT

The executable *spi_obs_point* defines time intervals with stable attitude during the observation.

More details can be found in the spi_obs_point User Manual, which can be downloaded from the following page

http://isdc.unige.ch/Support/spi/doc/doc.html

8.4 Energy bin definition and events binning

8.4.1 SPIBOUNDS

The executable *spibounds* is used to construct energy boundaries to be used for the binning of SPI event data.

Options include uniform linear or logarithmic spacing, as well as superimposing sub-regions (e.g. of finer spacing) over a global binning scheme such that you can investigate more carefully particular spectral regions (e.g., for spectral regions likely to contain line emission).

Table 7: *spibounds* parameters included into the main script

Name (main script)	Name (executable)	Type	Description
spibounds_nregions	nregions	integer	Number of regions for which distinct binning algorithms are to be defined. example: `4`
spibounds_regions	regions	string	The boundaries of the energy regions. example: "`20,40,100,1000,8000`"

spibounds_nbins	nbins	string	The number of bins per region. Negative numbers indicate logarithmically spaced bins. example: "`-10,-10,-9,-7`"

8.4.2 SPI_OBS_HIST

spi_obs_hist main purpose is generation of the event spectra that may be used for imaging, spectral or timing analysis, but in the same time it also derives Good Time Intervals and the Dead time for each pointing and pseudo detector.

More details can be found in the spi_obs_hist User Manual, which can be downloaded from the following page

http://isdc.unige.ch/Support/spi/doc/doc.html

8.4.3 SPI_ADD_SIM

spi_add_sim allows to add a simulated source into a real observation analysis. A source position and flux must be given as input parameters. *spi_add_sim* uses the IRF response to compute the detector counts expected for this source, and adds the counts to the data set. This program is useful to investigate crowding effects, by comparing outputs with inputs for the added source(s) in the presence of other real sources.

This program must be used as part of the ISDC pipeline as it needs an OG group with all the relevant information, such as the pointing pattern, the detector list etc. It also does not create any new data structure, but it add counts to the detector spectra data structure (SPI.-OBS.-DSP), updating this file.

At the moment only one source can be added at one time. However, several sources can be added just by running the program several times.

The program works only if the (image) analysis is carried out in one energy bin only. If there is more than one energy bin, *spi_add_sim* does not do anything, and the detector spectra will not be changed!

Table 8: *spi_add_sim* parameters included into the main script

Name (Main script)	Name (executable)	Type	Description
spi_add_sim_CoordSys	CoordSys	string	Coordinate system possible values: "`CELESTIAL`", "`GALACTIC` default: "`CELESTIAL`"
spi_add_sim_SrcLong	SrcLong	real	Longitude of simulated source (degree) default: `80.`
spi_add_sim_SrcLat	SrcLat	real	Latitude of simulated source (degree) default: `19.`
spi_add_sim_FluxScale	FluxScale	real	Simulated source flux (ph/cm^2/sec) default: `0.01`

8.5 Background model generation – SPI_OBS_BACK

spi_obs_back produces background models for imaging, timing and spectral analysis. The simplest case is to assume a background that does not change with time. The background then has only one free parameter per detector in the image analysis. The goal of the background generation is to provide a model for the time (and energy) variations in order to limit the number of background parameters in a more realistic imaging solution.

In the default GEDSAT mode the background is assumed to be proportional to the saturated detector trigger rate. Up to five different components may be included into background model.

The computed model background rates are averaged over the corresponding period of good times. These rates are then multiplied by the live time to produce the final background counts stored in the output data structure.

The resulting background rates detector spectra are formatted as the events detector spectra.

More details can be found in the spi_obs_back User Manual, which can be downloaded from the following page

http://isdc.unige.ch/Support/spi/doc/doc.html

Table 9: *spi_obs_back* parameters included into the main script

Name (Main Script)	Type	Description
spi_obs_back_nmodel	integer	Number of background model components possible values: 1 – 5 default: `1`
spi_obs_back_model0k	string	Model type of component **k** possible values: "`CONST`", "`ONTIME`", "`DEADTIME`", "`DEADLIVE`", "`ADJA CENT`", "`VETOGATE`", "`VETONONSAT`", "`VETOSAT`", "`GEDRATE`", "`GEDSAT`" default: "`GEDSAT`" ($k \neq 2$) "`ADJACENT`" (k=2)
spi_obs_back_mpar0k	string	Model parameters of component **k** default: ""
spi_obs_back_norm0k	string	Component **k** normalization type" possible values: "`NO`", "`GLOBAL`", "`DETE`", "`LINE`", "`CONT`", "`OFFLINE`", "`OFFCONT`" default: "`NO`"
spi_obs_back_npar0k	string	Component **k** normalization parameters default: ""
spi_obs_back_scale0k	real	Component **k** scaling factor default: 1.0

8.6 Image reconstruction and spectral extraction – SPIROS

A detailed description of *spiros* is given in the Spiros User Manual written by the author of *spiros* P.H.Connell, [6]. Here we give just some brief explanations and describe only the parameters used by main script. Note that even for the explained parameters we do not give the full range of possible values.

SPI is a coded-mask instrument, so direct deconvolution is in principle possible, but in practice the response is complex and the data include many pointing directions of the instrument, so 'indirect imaging' is essential. Indirect imaging implies 'forward-folding': for any candidate image we convolve with the instrument response function and compare the result with the observed data. This gives the basis for any iterative method which seeks to successively improve the agreement of the predicted with the observed data by adjusting the image. One approach is to regard the image as made up of 'point sources' and to adjust their positions and fluxes to give a best-fit to the data. This is the basic principle of the *spiros* method.

The primary purpose of *spiros* is to use SPI data to locate **point** or **point like** sources in the observation field of view and to output a catalog of their parameters along with images, their spectra and any flux variability in time.

spiros has three basic operating modes to choose from:

- **IMAGING** mode.
  Here the aim is only to locate point or point like sources in the observation field of view and to output a catalog of their locations and width parameters. This step also produces images.
- **IMAGING-TIMING**
  This step produces images in the case that variable bright sources are in the field of view.
- **SPECTRAL** and **SPECTRAL-TIMING** mode.
  In this mode *spiros* reconstructs spectra of sources in the SPI field of view. Source positions are either taken from the source-location process (IMAGING mode) or from a catalog. The output spectra can be used for model fitting in XSPEC. The SPECTRAL-TIMING mode is no different from the normal SPECTRAL mode except that it conside variable sources as in the timing mode
- **TIMING** mode.
  As in **SPECTRAL** mode a catalog of known sources is used to create a light curve of sources and background in several energy bins (one flux value per pointing).

8.6.1 IMAGING mode

Given its main task of locating point sources and creating images of them in the observation field of view the first question is how *spiros* goes about locating them.

A fast and simple method results by assuming some **prior knowledge** of the sources expected in the field of view, namely if they are **point** or **point like** and that there are only a **small finite** number of them, tens but not hundreds or thousands.

On this assumption the sources can be located using the **Iterative Removal of Sources** (**IROS**) method which literally attempts to locate each source in the field of view one after the other from the strongest to the weakest.

This method works as follows:

- A fast but blurred **location image** is created. It shows the most probable location or region of the greatest source emission. This image may be called a "pixel independent correlation

map" as it is constructed by scanning the imaging field of view with a "source probe" and calculating the flux by assuming all input counts are due to that source and any background events alone. This typically produces a "convolved" image with a global maximum at a good first guess for a probable source location.

- With this **approximate** location find the strength (probably upper limit), of the source and then the detector counts expected from it.
- Subtract source counts from the input count data to create **residue** count data more or less due to other sources not yet located.
- With these residue counts the procedure above is repeated creating a new location image, searching it for any new source and adding it to a growing list. Each time a new source is added to the list all sources are allowed to "float" to more precise locations and fluxes before their counts are subtracted from the input counts.

IROS repeats the iterative procedure above, locating each new source approximately, always allowing them to float to more optimal locations, creating new residue count data and stopping when nothing significant can be found. This procedure can be repeated in any number of energy bins to calculate a mean location for each source over the entire spectrum range.

For the user running *spiros* in its **IMAGING** mode the main task is to get the appropriate binned count data output from *spi_obs_hist* and decide how many sources to look for. The user may have an input catalog of known sources to get their values without calculating the position, or to find an extra one or two in addition, or just to make an image of them.

8.6.2 SPECTRAL extraction mode

The SPECTRAL mode of *spiros* is even easier to use as it does not search for sources but gets their locations and width from an input catalog, which is located under spi/source_cat.fits (in case the default names from the spi_science_analysis.par file have been applied). Sources, which have a SEL_FLAG=1 in this input catalog, will be taken into account. A different input catalog than the one attached to the Observation Group can also be given via source-cat-dol parameter. *spiros* finds the flux values of all of the flagged sources simultaneously but independently in each energy bin of the input observation count spectrum.

This method of calculating source flux values independently in each energy bin is required as a base line mode of *spiros* to output "raw" or photo peak spectra which are then to be used by the XSPEC spectral modeling software to extract their true spectra (this is described in the Cookbook Section 6).

8.6.3 TIMING analysis mode

The TIMING mode of *spiros* is again easy to use as it does not search for sources but gets their locations from an input catalog as in SPECTRAL mode. The idea is that the user will bin count data over some observation period in as many energy bins as desired and that *spiros* will calculate source flux values at a sequence of MJD times over some integration time interval to be given by the user. Note that the data at this level are already grouped into pointings, thus no timing analysis can be performed on time scales shorter than the pointing duration.

At present TIMING mode has three forms of analysis

- **QUICKLOOK**
  This computes the flux for all sources and background in small groups of pointing exposures covering the integration time interval, in parameter **source-timing-scale**, and in each count spectrum energy bin.

- **WINDOW**
  This mode assumes a mean count modulation model (MCM), or known background variations and computes source flux values simultaneously in a sequence of integration time intervals spanning the period of observation. For a large number of integration intervals and energy bins this might take some time to calculate.

- **TRANSIENT**
  This upgraded mode is similar to **WINDOW** mode but allows the removal of the discontinuous nature of the **HAT** like time bin functions to return an implicitly smoothed light curve. The integration time intervals spanning the period of observation are used here to create a sequence of **splining nodes**, or knot points, which will define a sequence of piecewise or smoothly continuous **B-spline functions**.
  Select either TRANSIENT-HAT, TRANSIENT-LINEAR, TRANSIENT-QUADR, or TRANSIENT-CUBIC. If no **-XXXX** function is appended a simple **HAT** type is selected and the results should be the same as in **WINDOW** mode.
  These functions are actually created by taking a HAT function covering each time interval and smoothing it with a HAT filter several times over to create LINEAR, QUADRATIC or CUBIC B-spline functions, a sequence which will tend towards a Gaussian shape with each further smoothing.

8.6.4 Parameter list

The *spiros* parameters included into the main script are given in the Table 10.

Table 10: *spiros* parameters included into the main script.

Name	Type	Description
mode	string	Select processing mode: possible values: "`IMAGING`", "IMAGING-TIMING", "`SPECTRA`", "SPECTRAL-TIMING", "`TIMING`", "`DIFFUSE`" default:"`IMAGING`"
background-method	integer	Handling method of background model response values possible values: `0` – ZERO, `1` – FIXED, `2` – FLOATING, `3` – FUNCTION, `4` – DET-GROUP `5` – MCM `6` – DP-modulation default:`5`
optistat	string	Enter optimization statistic possible values: "`CHI2`", "`LIKEH`" default:"`LIKEH`"
solution-constr	string	Image solution constraints possible values: "`NONE`", "`POSITIVE`" default:"`NONE`"
Selection Parameters		

energy-subset	string	Enter subset of energy bins [a-b,c-d,f,g.....]: default:“ ”
pointing-subset	string	Enter subset of pointing bins [a-b,c-d,f,g.....]: default:“ ”
detector-subset	string	Enter subset of detector bins [a-b,c-d,f,g.....]: default:“ ”
		spiros Timing Mode
source-timing-mode	string	Enter timing analysis mode possible values: “QUICKLOOK”, “WINDOW”, “TRANSIENT” default:“WINDOW ”
source-timing-scale	real	Enter timing observation integration time [days/seconds]: default:0.5
		General imaging options
nofsources	integer	Maximum number of sources to be searched for. default:3
sigmathres	real	Enter lower sigma threshold to stop source search: default:3.0
srclocbins	string	Enter count spectrum bins used for source location: possible values: “BLANC”, “SUM”, “FIRST”, “ALL” default:“FIRST”
reference-coord	string	Enter output image reference coordinate system: possible values: “RADEC”, “GALACTIC” default:“GALACTIC”
image-proj	string	Enter output image projection type: possible values: “CAR”, “TAN”, “AIT”, default:“AIT”
image-fov	string	Enter extent and location of output image: possible values: “POINTING+FCFOV”, “POINTING+ZCFOV”, “POINTING”, “USER” default:“POINTING+FCFOV”
image-orient	string	Enter output image orientation: possible values: “STANDARD”, “USER” default:“STANDARD”
image-pole-long	real	Enter longitude of output image pole [degrees]: default:0.0

image-pole-lat	real	Enter latitude of output image pole [degrees]: default:`90.0`
iteration-output	string	Enter whether output should be produced at each iteration default: "`NO`"
location-max-error	real	Maximum admissible input error source location default:`0.1`

8.7 Spectral Extraction – SPMODFIT

The tool *spimodfit* is intented to perform "model fitting" with SPI data. By "model fitting", we mean the fit of a linear combination of sky emission models, point sources and instrumental background. The coded-mask imaging γ-ray spectrometer SPI on the INTEGRAL space observatory will detect point sources and diffuse extended emission with an angular accuracy of about 1° over its energy range of 20–8000 keV. The purpose of the software package described here is to represent the measurement in terms of spatial models of the sky, fitting parameters of these models and estimating their uncertainty. This tool is oriented towards large-scale surveys, which combine a large set of individual pointings of the spacecraft. It concentrates on fitting spatial as opposed to spectral models, although (through using the different energy channels of the measurement) it will be an important method to generate spectra of diffuse emission.

The principal use will be fitting to maps of physically-based tracers of γ-ray emission. Examples are fitting the 1809 keV ^{26}Al line to free-free or IR emission maps, fitting diffuse continuum γ-ray s to gas (HI, CO) maps. It will also allow fitting spatial functions (e.g. Gaussian, exponential) where there is no 'physical' model available. Fitting components is the best method to get the spectra of diffuse emission when a good spatial (but a poor spectral) model is available. The analysis of measurements with this tool is complementary to deriving results from deconvolved images (e.g. from methods specifically designed for point sources (e.g. *spiros*). *However, it can be fruitfully used to extract the spectra of point sources.*

The algorithm performs fitting of raw data (binned counts for many observations) to multi-component models using the full instrument response information. In addition to parameter estimation, a Bayesian statistical analysis is used to include information on model characteristics and other prior information, so that the uncertainties of the fitted parameters are properly assessed. The prime results of the package are fitted model parameters with their uncertainty; complementing these, the fitted models are given also in the forms of skymaps and spectra.

This program is based on a former model fitting program for SPI, *spidiffit* developped by A. Strong (MPE, Garching). The main changes in *spimodfit* (w.r.t *spidiffit*) are:

- a more flexible handling of time variability (for all kind of sources)
- time-dependent instrument response function (IRF)
- the use of NAG-free constrained minimization procedures and the addition of a Levenberg-marquadt algorithm
- optimized matrix calculations and memory usage
- more output files (updated catalog of sources, background model, residues, etc.)

spimodfit is mainly dedicated to the study of long duration observations, large scale emission, focusing on the emission morphology.

For the details of the fitting methods and the statistical issues, we refer the reader to the handbook of spimodfit , from which this section of the handbook is extracted.

8.7.1 Use of *spimodfit*

From the user point of view, the data processing is done in 4 major steps, which all have their own options :

- Definition of the input observation data : on input, *spimodfit* expects the different files constituting a SPI observation (events rate, poitings, dead times, good times, energy bins and *optionally* background models). The energy bins can be redefined and will be processed independently. It is also possible to select a subset of the detectors in the input data. This step are usually performed via the standard OSA processing from GTI to Background levels, but ready to use data are also available at `http://www.mpe.mpg.de/gamma/instruments/integral/spi_public_service/data/`. These data are binned at high spectral resolution, therefore the user interested in point sources should rebin the energy channels using the parameters `first_energy_bin`, `last_energy_bin`, and `m_energy_rebin` in *spimodfit* . The background models can be collected into one single model setting the flag `spi_flatfield_single` to `yes` or limited to a maximum number (only the first `n_background_loaded` ones will be loaded). In addition to these files, the Instrument response function files IRFs file has to be specified. More than one IRFs files can be given to handle a time dependent response, the default values are appropriately inserted in the standard parameter file.
- Definition of the fitted parameters. *spimodfit* can fit extended emission ('images'), point sources ('sources') and background parameters. For each of these components, the time and detector variability should be given,as well as some other fitting or normalization parameters explained below.
- Fitting options, as the minimization routines and its initialization; MCMC options are currently not implemented.
- Definition of the output files. *spimodfit* generates a FITS file table containing the results of the fit. By default this file contains the different sky models (original map and fitted images) and a binary table containing the fitted parameters with associated configuration and estimated errors. Optionally, it can also add an exposure map in this FITS file, create a background model for the observation, write residues of the model, light curves, spectra and an updated point source catalog into other files.

8.7.2 spimodfit_analysis script

A set of standard parameters based on the ones used for the `SPI` quick look analysis available at `http://www.mpe.mpg.de/gamma/instruments/integral/spi/www/public_data/index_srcspe.html` is implemented in the script `spimodfit_analaysis` that is delivered for a more user friendly utilization of *spimodfit* .

The first steps before the background calculation are the same of `spi_science_analysis`. The energy channels of the input data and the set of detectors which are used must be coherently set between the binning and spectral extraction steps.

Regarding the background the default option is the calculation internal to *spimodfit* . The time variability is set by default to account for variability on science window time scale and at each annealing or detector failure. A feature which regards the "background multipliers" is introduced to account for different efficiency of the pixels. The relative default time variability accounts for the annealings and detector failures, and a monthly timescale variation to be fitted.

In case the user would like to use the external background, the parameter `background_input_file` should be set to `back_model_idx.fits[1]`. Either the flatfield, template or model options can be used. For the *flatfield* option, the option `spi_flatfield_single` should be set to `yes`, as done in the default settings. Also a combination of several background models can be used by setting the

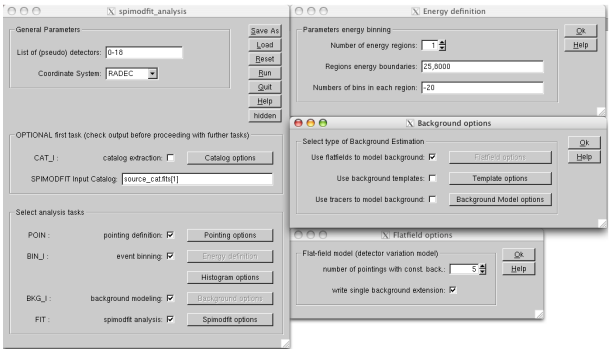


Figure 42: GUI with the standard settings for a flat-field background and logarithmically increasing energy binning.

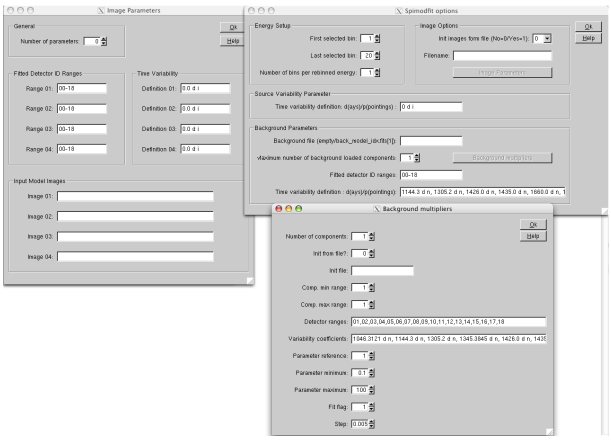


Figure 43: GUI with the standard settings for internal background computation, no image fitting and standard background multipliers.

appropriate parameters. For this option and the introduction of images for the diffuse emission models, we refer to the *spimodfit* handbook. However, we warn the reader that the full exploitation of these and more advanced features should be carried out in touch with the experts at *MPE*.[7]. In Fig. 42, we show the GUI with the standard settings for a flat-field background and logarithmically increasing energy binning.

Finally, the *spimodfit* final products are not attached to the observation group, decoupling the spectral extraction step from the previous ones. The program can then be used separately once the data are binned and the optional background is created, without the need of running any DAL tool to clean the observation group. However, the side effect is that the user should delete the output files by hand if he/she wants to run *spimodfit* another time since the program does not replace existing files. In Fig. 43, we show the GUI with the standard settings for internal background computation, no image fitting and standard background multipliers.

8.7.3 Use and syntax of the parameter file

spimodfit can be used from the command line after preparing the input data. This modality is suggested for advanced use. A script `spimodfit_rmfgen.csh` is created to call spirmf for the adopted energy binning. The script should be run after *spimodfit* from the command line.

[7] `http://www.mpe.mpg.de/gamma/integral/`

The parameters of the the program are defined in a file called *spimodfit* .par. This file should be readable in one of the directory set in the `PFILES` environment variable. Usually, the user updates the parameter file in the working directory, so one should set the variable accordingly : `setenv PFILES .`

The syntax of the file follows the IRAF standards (that is the one used by ISDC softwares).

In this file, each parameter is described in the format :

`parameter_name, parameter_type, parameter_mode, default_value, minimum_value, maximum_value, description_prompt`

The parameter_type can be "b" (boolean), "i" (integer), "r" (real), "s" (string) or "f" (filename). The "f" type can be followed by any combination of "r" (read access), "w" (write access), "e" (existence of file) "n" (absence of file). Thus, the "fw" type means to test whether the file given as a value of the parameter is writable.

The parameter_mode can be "a" (auto), "h" (hidden), "q" (query), "hl" (hidden+learn), "ql" (query+learn). With the "a" mode, the effective mode is set to the value of the parameter named as "mode" in the parameter file. If the "mode" parameter is not found, the effective mode is set to "h". With the "h" mode, no questions are asked, unless the default value is invalid for a given data type, whereas with the "q" mode you are always asked for a parameter. With the additional "l" mode, if an application changes a parameter value, the new value will be written to the parameter file when the application terminates.

spimodfit makes use of parameter types "i", "r" and "s".

Some of the parameters require to enter multiple arguments with the same parameters. This is done via a string which is then decomposed into different elements and interpreted. A multiple arguments parameter can be either a vector or a list :

- A vector is an array of values of the same kind (i.e. reals, integers, etc.), separated by blanks, for example : "1100.5 1433.0 1500.0" is a vector of 3 reals. This is a extension or the IRAF format coded in the OSA PIL library.
- A list is a array of values or fields (not necessary of the same kind) separated by commas. Sometimes an additional keyword can be put at the end of the list to modify its interpretation. For example : "1785-1802, 1802-1815, 1815-1826 keV" selects 3 energy ranges in keV. This possibility is a local extension of the IRAF standards.

Unless specified in the parameter description, all parameters are required to appear in the parameter file, even if their value is not used by the program.

Blank lines and lines beginning with '#' (comments) are ignored.

8.7.4 Suggested settings for point source analysis

The parameters, as done in the default parameter file, should be set in order to have:

- an isotropic background component.
- the contributions from those point sources which are within 20° from the optical axis of the instrument for a given science window.

The isotropic component is modelled as the product of a common scaling factor and a relative sensitivity for each of the 19 detectors of SPI (taking into account the three detector failures during the course of the mission so far). The relative detector sensitivity is permitted to vary on

a timescale of 10 revolutions (nearly 30 days) as a compromise between assuming no variability at all (resulting in a degraded fit) and the minimum reasonable variability timescale of one revolution (approx. 3 days, resulting in larger uncertainties due to the increased number of fit parameters). Assuming a shorter timescale than 30 days did only change the results within their statistical errors. The common background scaling factor was permitted to vary on a short timescale of 2 h (corresponding to typically four science windows). This was the conservative choice determined by the actually observed changes in the overall event rate (which is strongly dominated by background) and the aim to minimise the number of model parameters in order to improve sensitivity.

A default energy binning can be chosen based on considerations of SPI energy resolution and flux sensitivity. It consists of 20 energy bins of logarithmically increasing width between 25 and 8000 keV. with a narrower bin around the 511 keV pair annihilation line. Many of our sources of interest are too weak to be detected significantly above 200 keV in individual bins of even this coarse (compared to the SPI energy resolution) energy binning. So a wider binning consisting of 12 bins can be introduced for weaker sources. The narrow and the wide binning should have the first three bins (up to 48 keV) in common, then the wider binning roughly combines every two narrow bins up to 500 keV and after that every 3 narrow bins. The bin boundaries of the narrow binning could for instance be 25, 31, 39, 48, 60, 75, 93, 116, 144, 179, 223, 278, 346, 502, 520, 668, 832, 1036, 1290, 1606, 2000, and 8000 keV. The bin boundaries of the wider binning could be 25, 31, 39, 48, 70, 103, 150, 219, 320, 502, 520, 1000, and 8000 keV.

When the model is constructed, all parameters can be tested for relevance, and parameters which do not show an effect on the fit function can be excluded. Using a maximum likelihood fit which is initialised from a χ^2-fit, the model is then fit to the data and the statistical errors are determined from the covariance matrix. For more details see the *spimodfit* specific handbook. The entire analysis can be carried out independently in each energy bin such that boundary conditions such as variability timescales and the input point source catalog can in principle be chosen differently for each energy bin. This also allows for parallel computation. However, spectra should then be collected by the user with tools which should be developed *ad hoc*.

If the user should run again *spimodfit* outside the `spimodfit_analysis` chain, since its products are not attached to the observation group, he/she should manually delete the previous results, e.g. with the command: `rm spectr* back_model*spimodfit* res* fitquality.fits spimodfit_rmfgen.csh lightcurves.fits catalogue.spimodfit.fits`

8.7.5 Known issues

spimodfit handles time variability through the use of splines. The spline order can be 0 to 5, 0 corresponding to a piecewise constant function (with one scaling parameter per interval) and 3 corresponding to cubic splines. In many cases, when using n-order splines ($n \geq 1$), the fitting algorithm fails to find the optimal parameters. This is thought to be due to over-parametrized time variability because of the additional parameters of the splines.

Crashes have been reported when using a large $>\sim$ 1000 files dataset, their origin is still unclear.

8.8 Response interpolation – SPIRMF

spirmf performs a 2-dimensional interpolation of an existing RMF onto a desired grid in (input energy) x (output energy) channel space. This accommodates spectral analyses invoking essentially arbitrary binning of the data. The current limitations are that 1) the input and output matrices must be square (i.e. number of detector channels equals number of input energy bins), 2) the input matrices cannot be compressed (compression here referring to the OGIP condensed matrix file format), although the output matrices are automatically compressed, and 3) the output energy grid must fall within the span of the input grid (or very nearly so). *spirmf* can be used for either XSPEC-12 compatible component RMFs (see appendix B.0.3), or for single RMFs for use with

spiros extracted spectra. Elements above the diagonal are treated as zero. *spirmf* can optionally update a PHA-II data file to facilitate ease of use with XSPEC.

Both the input and output RMFs will be in the condensed file format, with zero-valued elements ignored.

8.9 Image reconstruction – SPISKYMAX

The detailed description of *spiskymax* is given in the document written by its author A.Strong, MPE Garching, [7]. Here we give only a brief overview of the program.

spiskymax regards the image as a pixelated skymap, and the aim is to obtain the intensity in each pixel. The most important idea is that the data constrain the image within some limits in an N-dimensional space, where N is the number of pixels. Hence there is no unique 'best' image and we have to make some choice out of all the possible images within the constrained region. Maximum Entropy method as implemented here is one way of doing this, and of quantifying the uncertainty of the result. Since the number of pixels is usually large (e.g. 10^4-10^6) the techniques involved are rather different from those of model-fitting.

spiskymax is adapted to the particular needs of SPI. The background temporal variations are treated via a template prepared by *spiback*, and the coefficients are fitted during the imaging process. The input count spectra contain data for many energy bins, and *spiskymax* analyses a subset (or all) of these energies as specified by the parameters *energy_range_min, energy_range_max*. The number of iterations can be specified since the automatic stopping criterion is not always appropriate and in any case may not be reached in the CPU time available. Sources to be analyzed are defined by their positions in the input source catalog (SPI.-SRCL-CAT); only those with the SEL_FLAG flag =1 are analyzed, and their fluxes and 1σ errors written to the output catalog.

After the image is produced, the flux and its error for a number of user-defined sources can be optionally determined. Each source is specified in terms of a position and 'ON' and 'OFF' radii. The ON region is the circle centered on the source position with radius ON and the OFF region is the annulus between the ON and OFF radii. The source flux is defined as

Σ(intensity in ON region)/Σ(intensity in OFF region) *solid angle(ON)/solid angle(OFF).

The error is based on a Bayesian analysis which marginalizes over all the unwanted degrees of freedom, so the significance of a source by this method may not correspond to a 'classical' significance level. In general significances increase for smaller skymaps since the degrees of freedom decrease.

8.10 Cleaning Tool - SPI_CLEAN

Run the SPI cleaning tool `spi_clean`, to remove all previous pipeline products:

```
spi_clean obs_group="og_spi.fits[1]" endlevel=PRP
```

Note that spi_clean is much faster than og_clean, but it only works on SPI data It is launched automatically by the by the pipeline when necessary.

spi_clean checks all entries in the OG (og_spi.fits). All entries that have a higher level than the requested end level are detached from the OG and deleted. If in force mode, *spi_clean* checks whether the detaching was successful, and if the file still exists, uses the UNIX rm command to delete file.

In force mode, *spi_clean* is also able to work with corrupted OGs (i.e., if one member of an OG has been manually deleted or renamed) and it will try to fix the OG. In this case, however, it cannot be guaranteed that the cleaning will work properly.

Table 11: *spi_clean* parameters

Name	Type	Description
obsgroup	string	DOL of observation group default: "`og_spi.fits[GROUPING]`"
endlevel	string	output level of the OG possible values:"`PRP, CAT_I, POIN, BIN_I, BKG_I, IMA, SPE`" default: "`PRP`"
chatter	integer	verbosity level possible values: 0 – 5 Chatter=5 is for debugging default: 3
force	boolean	Force removal of files possible values: `no` – *spi_clean* will neither work on corrupt OGs, nor delete any files using the system command rm. `yes` – *spi_clean* will also work on broken OGs and delete any detached file using rm. Furthermore, *spi_clean* will also try to repair a broken OG in the sense that it checks that links in the OG are ok. Corrupt links are removed. The level of the OG is set to the highest level of the children of the OG. Be careful! The results when working on broken OGs can be quite unpredictable. default: "`yes`"
clean_swg	boolean	Determines whether science window group should also be cleaned (not recommended). default: "`no`"

9 Additional tools

In this section we describe additional tools which may help you to analyze the results of your analysis.

9.1 SPI_SPECTRAL_DISPLAY

This script is displaying source spectra produced with *spiros*. The result appears in a separate ROOT display, where you have a possibility to zoom, write output into a postscript, and much more. You will see a menu by clicking on the right button of the mouse. Note that depending on the mouse position (near one of the axes, in the middle of the window, etc.) the menus will be different. Parameters of the script are given in Table 12. There are no hidden parameters in the script.

Table 12: *spi_spectral_display* parameters

Name	Type	Description
obs_group	string	DOL of the observation Group to work with default: "`og_spi.fits[GROUPING]`"
spec_dol	string	DOL of the spectrum to be displayed. You have not to set it when an observation group containing spectra is given. default: " "
ebounds_dol	string	DOL of the energy boundaries. You have not to set it when an observation group containing spectra is given. default: " "
main_title	string	Main title of the plot (appears only on the top of the display) default: "`Spectral plot`"
perkev	boolean	Flux units possible values: "`yes` " for displaying the spectra in counts/cm**2/sec/keV "`no`" for displaying the spectra in counts/cm**2/sec default: "`yes`"
loglog	integer	Defines the scaling mode for the X and Y axis possible values: `0` – linear/linear scaling `1` - log/linear scaling `2` - linear/log scaling `3` - log/log scaling Instead of using this parameter, you should use the functionality which is implemented in the ROOT display, i.e. set loglog = `0` and apply logarithmic scaling clicking the right mouse button above the displayed spectrum. default: `0`
chatter	integer	Chatter level. possible values: `0 -- 10` `0` = no screen dump information, `10` = maximal screen dump default: `1`

9.2 SPI_DSP_DISPLAY

This script is displaying detector spectra using root. You can use this on data which are of BIN_I level or higher. The maximum number of spectra displayed is 20. As in the case of *spi_spectral_display* result appears in a separate ROOT display, see discussion in Section 9.1. Parameters of the script are given in Table 12. There are no hidden parameters in the script.

Table 13: *spi_dsp_display* parameters

Name	Type	Description
obs_group	string	DOL of the observation Group to work with default: "`og_spi.fits[GROUPING]''`
dsp_dol	string	DOL of the detector spectrum. You have not to set it when an observation group containing spectra is given. default: " "
ebounds_dol	string	DOL of the energy boundaries. You have not to set it when an observation group containing spectra is given. default: " "
detector	integer	number of the detector to display the spectrum possible values: `0` – `141`, also you can use `-1` for all detectors, and `-2` for their sum. default: `-2`
pointing	integer	pointing number to be displayed possible values: `-1` for all pointings, and `-2` for their sum. default: `-2`
main_title	string	Main title of the plot (appears only on the top of the display) default: "`Detector spectrum`"
perkev	boolean	Flux units possible values: "`yes` " for displaying the spectra in counts/keV "`no`" for displaying the spectra in counts parameters **perkev** and **persec** can be used in any combination. default: `yes`
persec	boolean	Flux units possible values: "`yes` " for displaying the spectra in counts/sec "`no`" for displaying the spectra in counts parameters **perkev** and **persec** can be used in any combination. default: `yes`
chatter	integer	Chatter level. possible values: `0` – `10` `0` = no screen dump information, `10` = maximal screen dump default: `1`

10 Known limitations

1. SPI is a complex Gamma ray instrument almost always dominated by background contributions. The scientific validation of the SPI data analysis going on at ISDC and at different instrument team sites is as of today far from complete. Users are encouraged to look critically at any result obtained with the ISDC software, and to use external comparisons and simulations when possible. Spurious results can be derived, for example, when using a wrong set of parameters and/or an incorrect background modeling.

2. The SPI instrument is equipped with a Pulse Shape Analysis (PSD) electronic which carries out a parallel processing of the single detector events in order to provide additional information about their pulse shape. The PSD information was intended to help reducing the background. Unfortunately, the in-flight background conditions are such that even the best experts have failed to achieve significant improvements with the PSD. Consequently, all the PSD related processing is currently disabled in the analysis pipeline. PSD events are simply used as standard single events. The basic user choice is then to analyze only single (+PSD) events specifying detector list of 0-18 in the analysis, or to consider double and triple detector interaction with detectors 19-84.

3. Three instrumental responses are now included in our system, characterizing SPI before, between, and after, the detector 2 and 17 failures. Another will be available in the next OSA release to cope with the failure of detector 5. The spi_science_analysis pipeline cannot currently handle a time variable response. The easiest is to analysis the three possible cases independently (our software then selects automatically the appropriate response), and to combine the results later on. It is possible however, to analysis different mixtures of different data together using one of the three responses as they are not too different. Great care should be taken in this case anyway to avoid possible biases (see the Tips and Tricks section of our documentation). The spimodfit_analysis can instead handle multiple responses which are appropriately used during the data processing. The final response accounts for the modified responses accordingly.

4. The spiros imaging software is quite a complex tool with many different options and parameters. Not all possible cases have been fully scientifically validated. The best tested modes include "imaging" and "spectral extraction", and the more robust background method is number 2 combined with a time variation model proportional to the Germanium saturated events. Other modes such as "timing" and "spectral timing" and other background methods are being further tested and validated. The spimodfit software is an on-going project which has now reached a stable configuration, but not all the features have been scientifically tested.

5. At least in one case, a long (staring) pointing which is split up into several science windows in the ISDC system is not handled correctly in the SPI data analysis. It concerns: ScWs 008200220010.001 008200220020.001 008200220030.001. Only the first pointing is properly included in the analysis, while the subsequent ones are ignored. Please report if you find any other such cases.

References

[1] ISDC/OSA-INTRO Introduction to the *INTEGRAL* Data Analysis.
http://isdc.unige.ch/Soft/download/osa/osa_doc/prod/doc_tree.html

[2] ISDC/OSA–INST-GUIDE Installation Guide for the *INTEGRAL* Off-line Scientific Analysis.
http://isdc.unige.ch/Soft/download/osa/osa_sw/osa_sw-7.0/osa_inst_guide-2.0.4.pdf

[3] SPI observer's manual
http://astro.estec.esa.nl/Integral/AO5/AO5_SPI_om.pdf

[4] The INTEGRAL spectrometer SPI: performance of point-source data analysis, Dubath, P., Knödelseder, J., Skinner, G.K., et al. 2005, MNRAS, 357, 420
http://isdc.unige.ch/Instrument/spi/papers/mnr8675[420-428].pdf

[5] Jürgen Knödlseder web page devoted to SPI software.
http://www.cesr.fr/ jurgen/isdc/index.html

[6] Connell, P.H.
Documentation of the SPIROS software package for imaging, source location, spectral extraction and timing analysis using observation data from the INTEGRAL spectrometer SPI.
http://isdc.unige.ch/Instrument/spi/doc/manuals/SPIROS_documentation.ps

[7] Strong, A.
spiskymax User Manual and Explanatory Supplement.
http://isdc.unige.ch/Instrument/spi/doc/manuals/spiskymax.ps.gz

[8] Walter, R.; Favre, P.; Dubath, P. et al., 2003, A&A 411, L25

[9] Beckmann, V.
in the proceedings of 27^{th} Rencontres de Moriond, p.417, 2002. (astro-ph/0206506)

[10] SPI Compact Source Analysis – prepared by the SPI Team
SPI Complex Case Analysis – prepared by the SPI Tea